

**UNIVERSIDAD CARLOS III DE MADRID**  
**ESCUELA POLITECNICA SUPERIOR**

**INGENIERIA INDUSTRIAL**



**PROYECTO FIN DE CARRERA**

**CONTROL DE AMORTIGUADOR MAGNETO-  
REOLÓGICO CON MICROCONTROLADOR**

**Autor: Miguel Ángel Torres Corrochano**

**Tutor: Michael Victorio García Lorenz**

**Director: M. Jesús López Boada**

Título: Control de Amortiguador Magneto-reológico con Microcontrolador

Autor: Miguel Ángel Torres Corrochano

Tutor: Michael V. García Lorenz

Director: María Jesús López Boada

## EL TRIBUNAL

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día \_\_ de \_\_\_\_\_ de 20\_\_ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE



“El genio se compone del  
dos por ciento de talento y  
del noventa y ocho por ciento  
de perseverante aplicación”  
(Ludwig van Beethoven)

### AGRADECIMIENTOS

El final de esta carrera ha llegado, de aquella vez que me aventure a intentar perseguir un sueño, estudiando para conseguir convertirme en un ingeniero; parece que fue ayer y ya han pasado años.

Llegado este punto, este paso en la vida, he de recapacitar de todo lo vivido, aprendido y luchado. Y mi primera percepción es dar las gracias, primero a mis padres y familia, por dejarme luchar por mi sueño, por apoyarme, por luchar porque yo consiga mi sueño, teniendo fe en mí, ayudándome en todo lo que han podido; sacrificando parte de su vida por mí. También me viene a la mente allá en mi infancia, en mi paso por el colegio e instituto, el haber estado rodeado de maestros y profesores que me ayudaban a desarrollarme como persona, a descubrir mis valores, a apoyarme en mi educación, a tener esa paciencia y trabajo, y reiterarme una y otra vez que siguiera estudiando, que nunca me iba a arrepentir de ello por mas piedras que encontrara en el camino.

Por todos esos compañeros que he tenido en la universidad, que me han ayudado a enriquecerme como persona. Por todos esos momentos inolvidables, tanto tensos, por la presión de las asignaturas, como por los momentos de risas, bromas y anécdotas inolvidables que hemos tenido siendo alumnos. Por los miles de momentos vividos que serán inolvidables.

Agradezco también a todos los profesores que me he encontrado en la universidad, por haberme transmitido toda su sabiduría, por hacerme ver a mi mismo lo que soy capaz de hacer al conseguir superar todas las barreras y exámenes que nos ponen. También agradezco toda la paciencia que han tenido conmigo para ayudarme en mi aprendizaje, y su buena disposición a enseñarme. Y porque no, agradecido a la vida, por dejarme realizar lo que siempre he soñado ser.

Y mi último agradecimiento es para el profesor Michael García Lorenz, mi tutor del PFC, por haberme ayudado, haber tenido paciencia y haberme siempre atendido en cualquier duda o en cualquier momento en los cuales andaba perdido a la hora de seguir desarrollando el proyecto, habiéndome clarificado cualquier duda y explicado las veces que hiciera falta cualquier concepto que no terminaba de entender.

## RESUMEN

El presente proyecto final de carrera me ha sido propuesto por el profesor Michael García Lorenz, interesado en la tecnología magneto-reológica de los amortiguadores. La idea es realizar un control electrónico en lazo cerrado de dicho amortiguador.

Para conseguir realizar esta idea se parte de la base de usar un amortiguador magneto-reológico procedente de un modelo de coche de la marca Audi. El sistema se realizara con un sistema aparentemente sencillo de lazo cerrado alimentado por un microcontrolador con capacidad de trabajo en tiempo real, concretamente el microcontrolador utilizado es un ARM7 de Philips. El sistema conseguirá un control progresivo de la capacidad de absorción de oscilación sobre el amortiguador del cilindro, variando así las propiedades de amortiguación del sistema.

El sistema tiene carácter automático, ya que el mismo, mediante el programa desarrollado va modificando la resistencia a la compresión del fluido que está dentro del amortiguador. Todo esto lo va realizando de una manera “intuitiva” y progresiva mediante el carácter PID que le otorga parte del diseño del programa.

# ÍNDICE DEL PROYECTO

1. INTRODUCCION Y OBJETIVOS .....	10
1.1 ESTADO DE LA TECNICA .....	10
1.2 OBJETIVOS.....	13
1.3 PRESENTACION DEL PROYECTO.....	14
2. COMPONENTES.....	15
2.1 MATERIALES UTILIZADOS .....	15
2.2 EL MICROCONTROLADOR .....	16
2.2.1. INTRODUCCION A LOS MICROCONTROLADORES .....	16
2.2.2. MICROCONTROLADOR UTILIZADO.....	16
2.2.3. ARQUITECTURA Y TIEMPO REAL.....	17
2.2.4. COMUNICACIÓN.....	18
2.2.5 .ESQUEMA Y USO DEL MCB2100.....	19
2.3. CIRCUITO DE CONTROL AMORTIGUADOR MR .....	21
2.3.1. INTRODUCCION .....	21
2.3.2. ELEMENTOS DEL CIRCUITO DE CONTROL .....	22
2.3.3. FUNCIONAMIENTO DEL SISTEMA DE CONTROL.....	23
2.4AMORTIGUADOR MAGNETO- REOLÓGICO.....	25
2.4.1. INTRODUCCION .....	25
2.4.2. FUNCIONAMIENTO DE UN AMORTIGUADOR MR .....	26
2.4.3. ELECCION DEL AMORTIGUADOR MR.....	27
2.5. SOFTWARE DE DESARROLLO.....	28
2.5.1. ORCAD DISEÑO ELECTRONICO .....	29
2.5.2. KEIL UVISION PROGRAMACION.....	30
3. DISEÑO .....	31
3.1. INTRODUCCION .....	31
3.2. PARTE ELECTRONICA .....	31
3.2.1. EL MICROCONTROLADOR .....	32
3.2.2. CIRCUITO DE CONTROL.....	37
3.2.3. AMORTIGUADOR MR (CARACTERIZACION) .....	38
3.3. PROGRAMACION DEL MICROCONTROLADOR .....	58
3.3.1. IMPLEMENTACION DE LAS FUNCIONES EN EL MICROCONTROLADOR .....	58
3.3.2. DIAGRAMAS DE FLUJO.....	66

3.4 DISEÑO DE LA ELECTRONICA .....	69
3.4.1.INTRODUCCION .....	69
3.4.2.AJUSTE CIRCUITO DE POTENCIA .....	70
3.4.3. TRATAMIENTO DE LA SEÑAL.....	78
3.4.4. CONTROL PID (AJUSTE).....	83
3.4.5. AJUSTE ADC ANCHO DE PULSO .....	87
3.4.6. DISEÑO EN PCB.....	89
4. PUESTA A PUNTO Y ENSAYO.....	104
4.1.INTRODUCCION .....	104
4.2.PRUEBAS FUNCIONAMIENTO TEST ANALITICO .....	104
4.3. PRUEBAS FUNCIONAMIENTO PID .....	106
4.4. AJUSTE DEL SISTEMA (GANANCIA).....	107
5. CONCLUSIONES.....	108
6. BIBLIOGRAFIA .....	109
 <b>ANEXO 1: PROGRAMA DEL ARM7 .....</b>	<b>110</b>
<b>ANEXO 2: ESQUEMATICO DEL CIRCUITO ELECTRONICO.....</b>	<b>114</b>
<b>ANEXO 3: PRESUPUESTO .....</b>	<b>115</b>

## INDICE DE FIGURAS.

Figura 1: Materiales magneto-reológicos .....	11
Figura 2: Placa MCB2100 .....	18
Figura 3: Esquemáticos 1 .....	19
Figura 4: Esquemáticos 2 .....	20
Figura 5: Esquemático Placa Electrónica .....	22
Figura 6: Amplificador no inversor .....	23
Figura 7: Mosfet de potencia .....	23
Figura 8: Amortiguador Magneto-reológico .....	25
Figura 9: Funcionamiento Amortiguador .....	26
Figura 10: Logo ORCAD .....	29
Figura 11: Pspice .....	29
Figura 12: Layout .....	29
Figura 13: Keil U Vision 3 .....	30
Figura 14: MCB2100 y Ulink .....	32
Figura 15: MCB2100 con Placa Electrónica .....	32
Figura 16: Configuración debugger 1.....	33
Figura 17: Configuración debugger 2.....	34
Figura 18: Configuración debugger 3.....	34
Figura 19: Simulador de Keil .....	35
Figura 20: Configuración Debugger 4.....	35
Figura 21: Configuración Debugger 5.....	36
Figura 22: Depuración en tiempo real .....	36
Figura 23: Esquemático de Potencia .....	37
Figura 24: Simulación Señales .....	37
Figura 25: Hoja características amortiguador.....	38
Figura 26: Ensayos amortiguador.....	38
Figura 27: Graficas Amortiguador .....	39
Figura 28: Circuito L.....	40
Figura 29: Grafica Circuito L.....	40
Figura 30: Circuito L .....	41
Figura 31: Grafica circuito L .....	41
Figura 32: Circuito RL.....	42
Figura 33: Grafica circuito RL .....	42
Figura 34: Circuito RC.....	43
Figura 35: Grafica circuito RC .....	43
Figura 36: Grafica 2 circuito RC.....	44
Figura 37: Grafica 3 circuito RC.....	44
Figura 38: Grafica 4 circuito RC.....	45
Figura 39: Grafica 5 circuito RC.....	45
Figura 40: Grafica 6 circuito RC.....	46
Figura 41: Respuesta amortiguador en Intensidad .....	46
Figura 42: Circuito de Potencia .....	47
Figura 43:Grafica Circuito de Potencia .....	48
Figura 44:Circuito de potencia 2.....	48
Figura 45: Grafica circuito de potencia 2.....	49
Figura 46: Grafica 2 circuito de potencia 2 .....	49
Figura 47: Circuito de Potencia 3 .....	50
Figura 48: Grafica circuito de potencia 3.....	50
Figura 49: Hoja de características del amortiguador.....	51



Figura 50: Circuito de potencia 4 .....	51
Figura 51: Grafica circuito de potencia 4 .....	52
Figura 52: Valores Intensidad amortiguador .....	52
Figura 53: Electrónica control amortiguador .....	53
Figura 54: Circuito de potencia 5 .....	54
Figura 55: Grafica circuito de potencia 5 .....	54
Figura 56: Grafica 2 circuito de potencia 5 .....	55
Figura 57: Circuito de potencia 6 .....	55
Figura 58: Grafica circuito de potencia 6 .....	56
Figura 59: Grafica 2 circuito de potencia 6 .....	56
Figura 60: Grafica 3 circuito de potencia 6 .....	57
Figura 61: Registro hexadecimal .....	59
Figura 62: Tabla conversiones .....	59
Figura 63: tabla configuración ARM7 .....	59
Figura 64: Ejemplo hexadecimal .....	60
Figura 65: PWM generado .....	62
Figura 66: Simulación señales programa .....	62
Figura 67: Captura del ADC .....	64
Figura 68: Lazo abierto .....	66
Figura 69: Lazo cerrado .....	66
Figura 70: Diagrama flujo programa principal .....	67
Figura 71: Diagrama flujo interrupción ADC .....	67
Figura 72: Diagrama flujo interrupción Timer 0 .....	68
Figura 73: Divisor de tensión .....	69
Figura 74: Filtro RC .....	70
Figura 75: Diseño Operacional .....	71
Figura 76: Barrido 1 Khz: Anchos de pulso (10%, 30%, 60%, 80%, 100%) Sin filtro .....	73
Figura 77: Barrido 1 Khz: Anchos de pulso (10%, 30%, 60%, 80%, 100%) Con filtro .....	73
Figura 78: Barrido 1 Khz: Anchos de pulso (10%, 30%, 50%, 70%, 90%) Con filtro. Señal Amplificada .....	74
Figura 79: Barrido Ancho de pulso al 50% con frecuencias de 1 Khz, 3 Khz, 7Khz, 15 Khz, 20 Khz. Sin filtro .....	74
Figura 80: Barrido Ancho de pulso al 50% con frecuencias de 1 Khz, 3 Khz, 7Khz, 15 Khz, 20 Khz. Con filtro .....	75
Figura 81: Barrido Ancho de pulso al 50% con frecuencias de 1 Khz, 3 Khz, 7Khz, 15 Khz, 20 Khz. Con filtro y Amplificada .....	75
Figura 82: Impedancia variable .....	76
Figura 83: Impedancia variable esquemáticos .....	77
Figura 84: Barrido Ancho de pulso al 50% con frecuencias de 1 Khz, 3 Khz, 7Khz, 15 Khz. Sin filtro .....	77
Figura 85: Barrido Ancho de pulso al 50% con frecuencias de 1 Khz, 3 Khz, 7Khz, 15 Khz. Con filtro .....	78
Figura 86: Barrido Ancho de pulso al 50% con frecuencias de 1 Khz, 3 Khz, 7Khz, 15 Khz. Con filtro y Amplificada .....	78
Figura 87: Simulación Circuito control potencia .....	79
Figura 88: Duty cycle 10% con frecuencias de 1 Khz, 3 Khz, 7Khz y 15 Khz. ....	80
Figura 89: Duty cycle 20 % con frecuencias de 1 Khz, 3 Khz, 7Khz y 15 Khz. ....	80

Figura 90: Duty cycle 40 % con frecuencias de 1 Khz, 3 Khz, 7Khz y 15 Khz .....	81
Figura 91: Duty cycle 80% con frecuencias de 1 Khz, 3 Khz, 7Khz y 15 Khz. ....	81
Figura 92: Esquema potencia mejorado .....	82
Figura 93: Señal PWM mejorada .....	82
Figura 94: Lazo cerrado PID .....	83
Figura 95: Respuestas PID .....	83
Figura 96: Configuración 1 .....	89
Figura 97: Esquemático PCB .....	90
Figura 98: Configuración 2 .....	90
Figura 99: Configuración 3 .....	91
Figura 100: Configuración 4 .....	92
Figura 101: Configuración 5 .....	93
Figura 102: Configuración 6 .....	94
Figura 103: Configuración 7 .....	94
Figura 104: AutoECO .....	95
Figura 105: Footprint .....	96
Figura 106: nombre footprint .....	96
Figura 107: huellas componentes .....	97
Figura 108: definir nodo .....	97
Figura 109: definición huella .....	98
Figura 110: guardar huella .....	98
Figura 111: huellas PCB .....	99
Figura 112: Capas de pistas .....	99
Figura 113: Creación ruteo pistas .....	100
Figura 114: Placa PCB .....	100
Figura 115: Check design rules .....	101
Figura 116: Corrección ruteado placa .....	101
Figura 117: Modificación grosores de pista .....	102
Figura 118: Ruteo acabado placa .....	102
Figura 119: Placa trabajando en lazo abierto .....	103
Figura 120: Depurando programa en lazo abierto .....	104
Figura 121: Placa trabajando en lazo cerrado .....	105
Figura 122: Depurando programa en lazo cerrado .....	106
Figura 123: Probando sistema en funcionamiento .....	107

## 1. INTRODUCCION

En este primer capítulo se presenta el desarrollo del proyecto, así como los objetivos fijados en él y una breve descripción de la tecnología utilizada y su incursión en el mercado.

### 1.1. ESTADO DE LA TECNICA

Desde antaño, el ser humano siempre ha gustado de desplazarse a conocer nuevas fronteras, experiencias, se puede decir, que es un ser nómada, incitado sobre todo por la inteligencia y la curiosidad por todo que esta le despierta. En la necesidad de desplazarse, el ser humano ha querido y conseguido ir mejorando esa capacidad de translación a lugares diferentes, ideando por ello cada vez aparatos más complejos. Y es por todo esto que a día de hoy nace lo que denominamos como “transporte”, se puede decir de ello, que es la capacidad que tiene el ser humano de desplazarse tanto a él, como a materiales o animales propios hacia lugares donde desea ir. Nosotros concretamente nos centramos en la capacidad de transporte del propio ser humano. En ese medio de transporte, que utiliza para ello llamado turismo, y más concretamente en un aspecto de él, su seguridad, centrándonos en una parte de su seguridad que es la estabilidad en carretera.

Hoy en día, el sector automovilístico está muy desarrollado tecnológicamente. Es notable antes de nada estudiar el objetivo de desarrollo del automóvil y más concretamente en España. Se puede decir, que en la década de los noventa, hasta bien entrados la década del nuevo milenio, el desarrollo estaba centrado en la potencia y prestaciones de velocidad y fuerza que podía desarrollar un vehículo. Sucede que a mediados de la década pasada, hubo un cambio de conciencia y la gente comenzó a demandar más seguridad y confort en los vehículos y no tanto una gran potencia desarrollable en el coche. Hoy en día podemos decir que se unen potencia, seguridad y confort. Y es en el apartado de seguridad, donde nos centramos, esa variable que para que se cumpla debe ser capaz de controlar los dos parámetros anteriores, potencia y confort. Potencia, porque debe ser consciente de que se tiene que controlar y ajustarse a sus características y confort, porque esto es lo que debe percibir el conductor y sus ocupantes para tener la sensación de seguridad.

En el apartado de la seguridad, tenemos varios puntos o factores a tener en cuenta. Nosotros nos centraremos como ya hemos comentado más atrás, en el aspecto de la estabilidad del coche ante los diferentes trayectos que puede realizar en su vida útil.

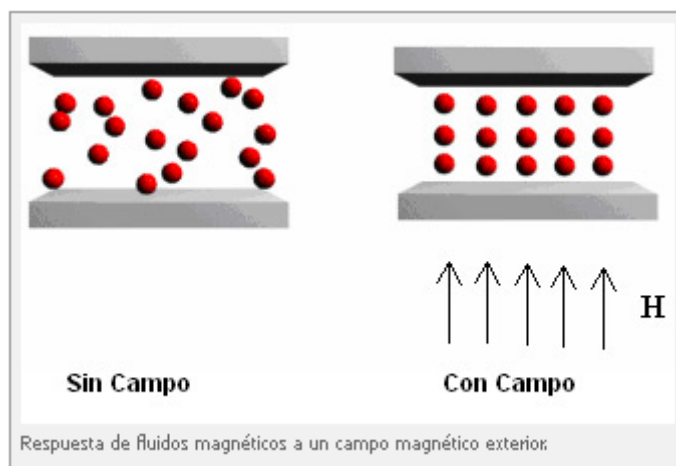
En el estudio de la estabilidad tiene una gran repercusión la amortiguación del coche, así como su comportamiento en todos los tipos de terrenos, es por ello que es primordial que esto actué correctamente en cada ámbito descrito.

En el mercado actual, y en la industria, podemos decir que la tecnología ha avanzado mucho, y ha tenido un gran salto debido al descubrimiento de los materiales magneto-reológico, creando con ellos los denominados amortiguadores magneto-reológico, llamados por algunos también como “amortiguadores progresivos”.

Podemos hacer una reseña para explicar cómo funcionan estos materiales magneto-reológicos. Los materiales magneto-reológico (MM), son materiales que responden a la aplicación de un campo magnético con un cambio en su comportamiento reológico y están formados por partículas magnetizables finamente divididas y suspendidas en un líquido portador, tal como aceite mineral, keroseno, etc. o en un sólido portador con elasticidad suficiente para permitir la orientación de los dipolos ante el campo magnético externo.

La respuesta producida en los materiales magneto-reológicos, es el resultado de la polarización inducida en las partículas suspendidas mediante la aplicación de un campo externo. La interacción entre los dipolos inducidos resultantes, obliga a las partículas a formar estructuras en forma de columna, paralelas al campo aplicado.

Este tipo de estructuras forman una estructura similar a una cadena, la cual restringe el movimiento del fluido dando lugar a un incremento en las características viscosas de la suspensión. La energía mecánica necesaria para producir estas estructuras tipo cadena se incrementa conforme se aumenta el campo aplicado, produciéndose un esfuerzo dependiente del campo. En ausencia del campo aplicado, los fluidos magneto-reológicos muestran un comportamiento Newtoniano.



Un fluido se define como una sustancia o medio continuo que se deforma continuamente en el tiempo ante la aplicación de una sollicitación o esfuerzo cortante sin importar la magnitud de ésta. También se puede definir como aquella sustancia que, debido a su poca cohesión intermolecular, carece de forma propia y adopta la forma del recipiente que lo contiene.

Figura 1: Materiales magneto-reologicos

Se puede decir con este dibujo que este es el funcionamiento básico de estos materiales magneto-reológicos. [actimat.es / materiales magneto-reológico]

La principal compañía que a día de hoy desarrolla este producto de manera puntera es Delphi. La compañía Delphi, es una filial de la General Motors.

Audi está utilizando en la actualidad este método de suspensión en varios modelos de gama alta como el nuevo audi R8.

Audi MageneticRide se basa en un principio magneto-reológico. Un fluido magneto-reológico es muy especial ya que mediante la aplicación de un campo magnético podemos cambiar su comportamiento reológico, es decir cambiamos el punto de fluidez del material haciéndolo que se vuelva un semisólido, esto se puede hacer debido a que el fluido está compuesto por partículas magnetizan tés suspendidas en un liquido

portador que al polarizarse las partículas se alinean con el campo produciendo una estructura tipo cadena que cambia su composición final.

En un campo magnético, las nano partículas de hierro suspendidas en el fluido se alinean en la dirección del fluido magnético. La bobina electromagnética se integra en el pistón de amortiguación de tal forma que cuando recibe energía el fluido magnético se mueve transversalmente a los puertos de admisión en el pistón amortiguador. Si el pistón se mueve, las partículas de hierro alineadas crean una resistencia al flujo en el fluido de suspensión. La energía se controla en relación con la dinámica de conducción y los impulsos desde la carretera, consiguiéndose una situación óptima de amortiguación. Ello produce según el deseo del fabricante un tacto más confortable o un manejo más deportivo. Al no tener que mover partes mecánicas, la tecnología Delphi responde de forma extremadamente espontánea. La fuerza de amortiguación sólo depende de la potencia aplicada al fluido magneto-reológico y se puede ajustar hasta 1,000 veces en un segundo gracias al control variable de la potencia. Ello significa que los amortiguadores magneto-reológicos responden casi en tiempo real con gran sensibilidad a los impulsos desde la carretera. El control del sistema cerrado con los algoritmos adecuados asegura el mayor contacto posible entre la rueda y la carretera sobre superficies irregulares con los menores impulsos a la carrocería. [Alberto Rosa/Nano materiales y nano dispositivos]

### 1.2. OBJETIVOS DEL PROYECTO

El objetivo principal del presente proyecto ha sido el desarrollo del hardware y software de un sistema autónomo de reducido tamaño y bajo coste, capaz de controlar progresivamente la dureza y capacidad de amortiguamiento de este amortiguador magneto-reológico. La idea parte de que para que se consiga este control se haga por técnicas de ingeniería de control, como puede ser el regulador PID.

Las principales funciones que deberá cumplir el control diseñado para este amortiguador será:

- Comprobar continuamente, mediante mediciones de tensión el nivel de tensión que tiene el punto de referencia al que se quiere llegar.
- Mantener en un cierto nivel continuo todo el sistema, con variaciones pequeñas ante los grandes contrastes que se encuentre al actuar el amortiguador.
- Hacer un control progresivo mediante la actuación del regulador PID cada vez que se desee modificar el punto de referencia.

### 1.3. PRESENTACION DEL PROYECTO

El desarrollo de este proyecto ha sido propuesto por el profesor Don Michael V. García Lorenz para el estudio y aplicación de la tecnología de materiales magneto-reológicos.

En el presente proyecto se presenta la tecnología utilizada, el sistema y diseño con el cual se va a controlar, así como el comportamiento que tendrá una vez puesto en funcionamiento el sistema con el amortiguador, como también los estudios y ámbitos de aplicación que tendrá.

- La primera parte consistió en el estudio del amortiguador magneto-reológico. Estudio de las graficas del fabricante, modo de uso etc.
- La segunda parte consistió en el diseño de control del amortiguador magneto-reológico. Una vez estudiado el comportamiento y funcionamiento de este se procede a diseñar un sistema adecuado de control para el amortiguador mediante técnica de regulación y control, como son el lazo cerrado.
- La tercera parte consistió en el diseño electrónico de todo el sistema. Una vez diseñado en folio todo el sistema en lazo cerrado, se procede a traducirlo en diseño y componentes electrónicos.
- La cuarta parte consistió en la simulación mediante herramientas CAD del comportamiento del sistema de control electrónico para eliminar ruidos y simular posibles entradas y salidas que podría tener el sistema. Mediante el uso del paquete de herramientas CAD de ORCAD, simule distintas señales a distintas frecuencias con distintos pulsos para hacer estudio de posibles filtros para eliminar ruidos indeseables en el sistema.
- La quinta parte consistió en el diseñar el programa para el microcontrolador ARM7 para gobernar el sistema de control electrónico diseñado. Básicamente consistía en control del PWM mediante un bucle y un sistema PID adaptado al sistema.
- La sexta parte consistió en el diseño físico de la tarjeta de control electrónico diseñada anteriormente, ajustando tamaños y encapsulados al entorno donde se debería insertar. Para ello hice mediciones del entorno, así como busque en varias webs como son farnell, amidata, para ver los encapsulados de los componentes y proceder también a su compra.
- La séptima parte consiste en introducir el programa al ARM7, conectarlo con el sistema de control y el amortiguador y observar resultados. Puesta a punto de todo el diseño y ensamblado de todas las partes que componen este proyecto.
- La octava parte consiste en anotar resultados experimentales, obtener la respuesta que estamos buscando y buscar posibles aplicaciones.

## 2. COMPONENTES

### 2.1. MATERIAL UTILIZADO

En este capítulo se procederá a describir los componentes que han sido utilizados para el desarrollo del sistema de control del amortiguador magneto-reológico. Se puede separar todo el desarrollo en dos grandes pilares, el primero es la lógica y programación que lleva el control de este sistema, y el segundo es el diseño y desarrollo electrónico que se ha creado para este sistema.

Esquemáticamente se podría describir como:

1. Sistema de lógica

- Placa Keil MBC2100
- Microcontrolador ARM7 LPC2129

2. Sistema de control

- Placa control electrónico mediante Intensidad de señal
- Amortiguador Magneto-reológico



## 2.2. EL MICROCONTROLADOR

### 2.2.1 INTRODUCCION A LOS MICROCONTROLADORES

Podemos definir un microcontrolador como un circuito integrado programable que contiene todos los componentes de un computador. Se emplea para controlar el funcionamiento de una tarea determinada y, debido a su reducido tamaño, suele ir incorporado en el propio dispositivo al que gobierna. Esta última característica es la que confiere la denominación de “controlador incrustado”.

El microcontrolador es un computador “dedicado”. En su memoria solo reside un programa destinado a gobernar una aplicación determinada; sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar, y todos los recursos complementarios disponibles tienen como única finalidad atender sus requerimientos. Una vez programado y configurado el microcontrolador solamente sirve para gobernar la tarea asignada.

“Un microcontrolador es un computador completo, aunque de limitadas prestaciones, que está contenido en el chip de un circuito integrado y se destina a gobernar una sola tarea.”

### 2.2.2 MICROCONTROLADOR UTILIZADO

El microcontrolador con el cual se desarrollara este sistema es un microcontrolador preparador para procesar en tiempo real con un diseño especial el cual le permite tener una fluidez y rapidez en respuesta ante las entradas generadas. El microcontrolador se trata de un NXP LPC2129 (antes Philips) con arquitectura RISC.

El control y la programación la realizamos en lenguaje C mediante el software KEIL  $\mu$ Vision, diseñado especialmente para este microcontrolador, el cual nos permite una programación más sencilla que si tuviéramos que tratarlo y programarlo mediante lenguaje maquina o ensamblador directamente.

Los recursos que utilizaremos del microcontrolador serán los siguientes:

- Ancho de pulsos o PWM para variar la intensidad en el sistema y como consecuencia el nivel de tensión que amplifica el sistema para realimentarlo en el lazo
- Convertidor ADC que transforma el nivel de tensión recibido en valores digitales proporcionales al nivel analógico recibido.
- Timer para contar los ciclos y los cambios de ciclo, así como la detección de flancos en la señal.
- El reloj interno para controlar y sincronizar todo el sistema
- Los vectores de interrupción para realizar correctamente cada ejecución en el tiempo debido

### 2.2.3. ARQUITECTURA Y TIEMPO REAL

Este micro controlador posee arquitectura RISC, cuyo significado es "Computador de Juego de Instrucciones Reducidos". No solo implica que el número de instrucciones máquina que es capaz de interpretar y ejecutar el procesador es pequeño, sino también que posee las siguientes características:

1º. Las instrucciones son simples y rápidas

La falta de complejidad en la operación que realizan las instrucciones de los procesadores RISC permite que sean ejecutadas, mayoritariamente, en un solo ciclo de instrucción.

2º. Las instrucciones son ortogonales

Apenas tienen restricciones en el uso de operandos. Cualquier instrucción puede usar cualquier operando.

3º La longitud de las instrucciones y los datos es constante.

Todas las instrucciones tienen la misma longitud.

Este microcontrolador es capaz de operar en tiempo real, ahora definamos que es un sistema en tiempo real:

Un sistema en tiempo real (STR) es aquel sistema digital que interactúa activamente con un entorno con dinámica conocida en relación con sus entradas, salidas y restricciones temporales, para darle un correcto funcionamiento de acuerdo con los conceptos de predictibilidad, estabilidad, controlabilidad y alcanzabilidad.

La principal característica que distingue a los STR de otros tipos de sistemas es el tiempo de interacción. Sin embargo, antes de continuar es necesario aclarar el significado de las palabras "tiempo" y "real". La palabra "tiempo" significa que el correcto funcionamiento de un sistema depende no sólo del resultado lógico que devuelve la computadora, también depende del tiempo en que se produce ese resultado. La palabra "real" quiere decir que la reacción de un sistema a eventos externos debe ocurrir durante su evolución. Como una consecuencia, el tiempo del sistema (tiempo interno) debe ser medido usando la misma escala con que se mide el tiempo del ambiente controlado (tiempo externo). [wikipedia.org / sistemas en tiempo real]

Un STR tiene tres condiciones básicas:

- Interactúa con el mundo real (proceso físico),
- emite respuestas correctas y
- cumple restricciones temporales.

De esta forma hemos dado una descripción bastante buena de cómo está diseñado y cuál es el comportamiento del microcontrolador utilizado.

## 2.2.4. COMUNICACION

Nuestro sistema, en el cual está integrado el ARM7 que usamos como controlador está compuesto por los siguientes elementos y filosofía de comunicación:

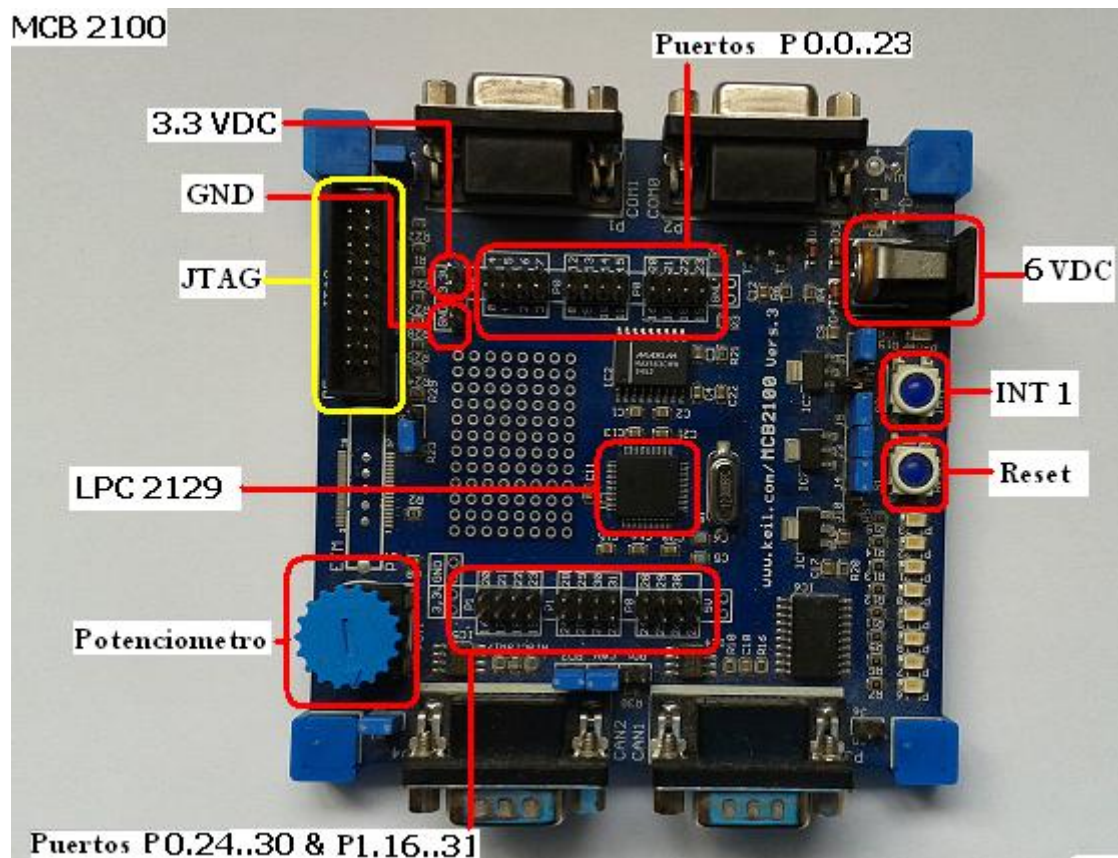


Figura 2: Placa MCB2100

### JTAG

Es la comunicación o puerto de comunicación que utilizara para conectarse nuestro depurador de código a la placa, que este a su vez se conectara directamente al PC mediante una conexión USB.

### 3,3VDC

Conexión de alimentación mediante la cual se conectara el sistema lógico de la placa electrónica del amortiguador magneto-reológico.

### 6 VDC

Conexión de alimentación mediante la cual se alimenta la placa de una fuente de tensión externa para que esta funcione correctamente.

## Puertos

Entradas y salidas de los pines mediante los cuales se conectarán el ARM a la placa electrónica para crear la conexión y comunicación de la placa de control del amortiguador magneto-reológico con el micro controlador ARM7.

### 2.2.5. ESQUEMA Y USO DEL MCB2100

Ahora estudiamos el esquemático de la placa y resaltamos las partes que van a ser utilizadas en el proyecto.

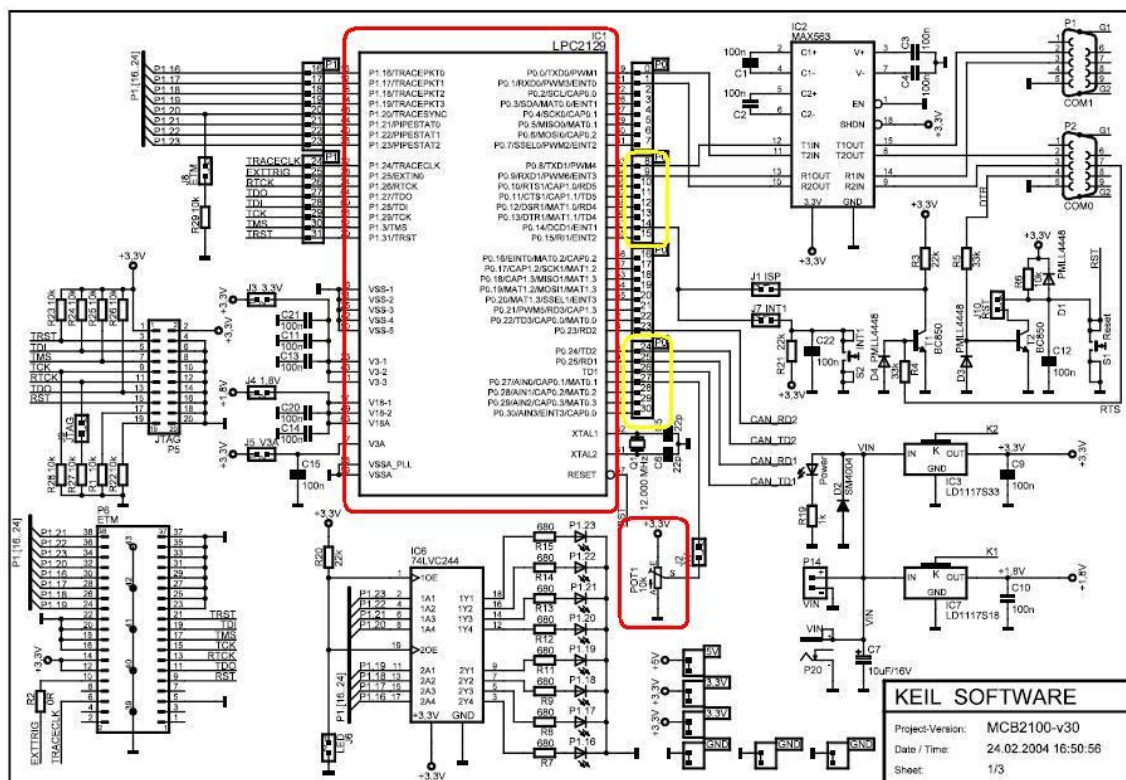


Figura 3: Esquemáticos 1

En este primer esquemático podemos observar todo el diseño del MCB2100, en cual se ha resaltado el micro controlador LPC2129, el potenciómetro y los dos bloques de puertos principales que vamos a utilizar para el control del amortiguador magneto-reológico.

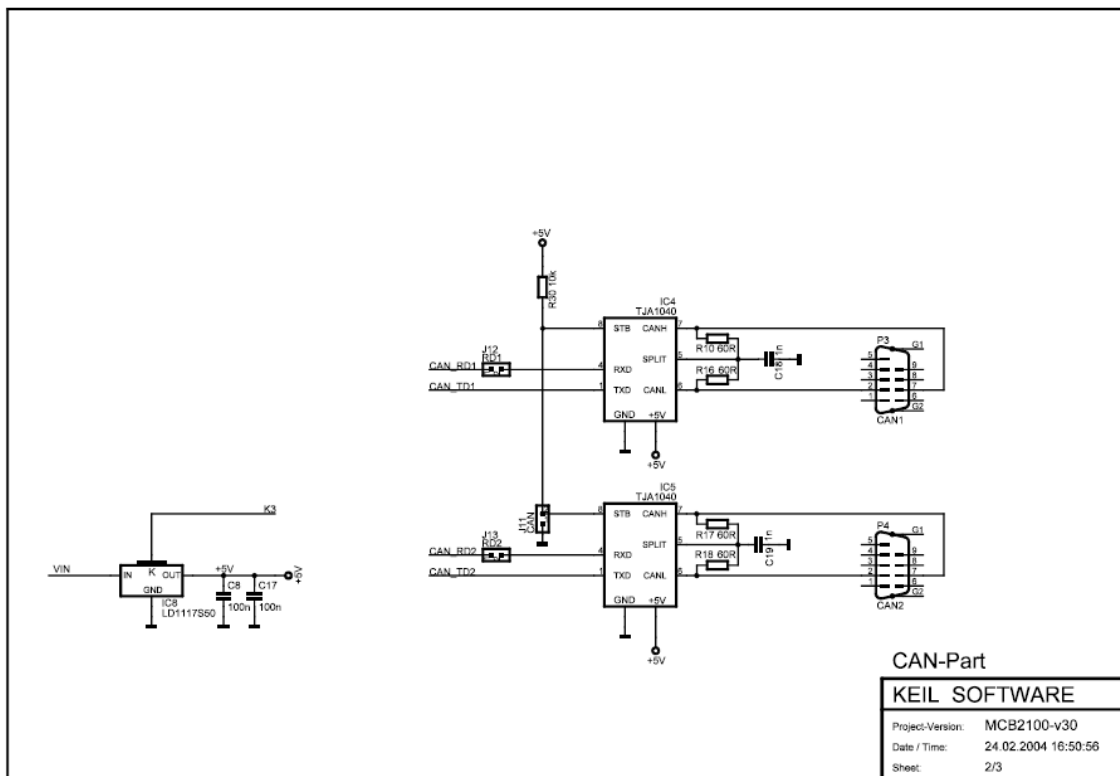


Figura 4: Esquemáticos 2

Este es el segundo esquemático donde se refleja la parte de diseño que quedaba sin resaltar de manera implícita en el anterior esquemático.

### 2.3. CIRCUITO DE CONTROL AMORTIGUADOR MR

#### 2.3.1. INTRODUCCION

El diseño del circuito de control del amortiguador magneto-reológico es sencillo y dinámico.

Este sistema está basado en el control de la intensidad que le llega al amortiguador, de este modo podremos obtener de manera precisa que rigidez presentara el amortiguador a la hora de adsorber los obstáculos que se pueden presentar en el funcionamiento del mismo.

El principio de control en el cual se basa es en la generación de un tren de pulsos de intensidad generado por el transistor mosfet que nosotros controlaremos, que se traducirá en un tren de pulsos por tensión que le llegara como referencia al sistema y este amplificara la señal para mandársela al ARM7, el cual la interpretara y mediante el bucle de control y el sistema PID regulara el caudal hasta el punto deseado.

[illegible]

Página 22



### 2.3.3. FUNCIONAMIENTO DEL SISTEMA DE CONTROL

El sistema de control se sustenta en dos pilares básicos, el control y generación del ancho de pulso y la amplificación de la señal para tratarla.

Para amplificar la señal hemos diseñado un control mediante un amplificador operacional no inversor, el cual su esquema sería el siguiente:

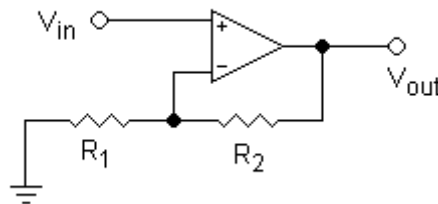


Figura 6: Amplificador no inversor

Y se basa en la ecuación matemática siguiente:

$$V_{out} = V_{in} \left( 1 + \frac{R_2}{R_1} \right)$$

De la cual se puede fácilmente deducir que la ganancia en amplificación de señal depende de R1 y R2, de este modo mediante nuestro potenciómetro podemos variar la ganancia del sistema para conseguir ajustar el funcionamiento de nuestro sistema.

Para generar nuestro tren de pulsos o PWM usamos un transistor mosfet:

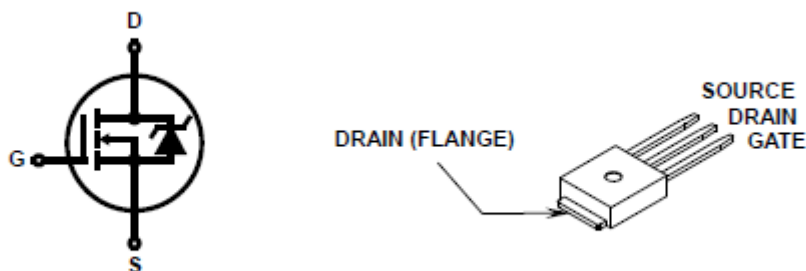


Figura 7: Mosfet de potencia

En la figura de arriba se presenta el esquema de representación del mosfet, así como el aspecto que presenta como componente en la tarjeta de control.



Podemos definir un mosfet por o siguiente:

El transistor de efecto de campo metal-óxido-semiconductor o MOSFET (en inglés Metal-oxide-semiconductor Field-effect transistor) es un transistor utilizado para amplificar o conmutar señales electrónicas. Aunque el MOSFET es un dispositivo de cuatro terminales llamadas surtidor (S), drenador (D), compuerta (G) y sustrato (B), el sustrato generalmente está conectado internamente a la terminal del surtidor, y por este motivo se pueden encontrar dispositivos de tres terminales similares a otros transistores de efecto de campo. El transistor MOSFET está basado en la estructura MOS. Es el transistor más utilizado en la industria microelectrónica, ya sea en circuitos analógicos o digitales, aunque el transistor de unión bipolar fue mucho más popular en otro tiempo. Prácticamente la totalidad de los microprocesadores comerciales están basados en transistores MOSFET. [wikipedia.org / Mosfet]

## 2.4. AMORTIGUADOR MAGNETOREOLOGICO

### 2.4.1. INTRODUCCION

Los amortiguadores magneto-reológicos usan la tecnología innovadora del uso de los materiales magneto-reológicos, de los cuales ya hicimos una reseña al principio de esta memoria.

La característica o ventaja principal que presentan estos amortiguadores frente a los amortiguadores convencionales es que podemos controlar la fuerza o resistencia que opondrán estos amortiguadores a comprimirse con solo hacer pasar corriente sobre ellos, creando así un campo magnético que influirá sobre el funcionamiento del fluido.

Esto nos supone una gran ventaja, ya que podremos obtener un amortiguador de unas características diferentes a cada cambio de nivel de intensidad que se produzca en el mismo, lo cual nos permite crear un amortiguador adaptativo a los diferentes entornos en los cuales se puede encontrar un sistema de suspensión.

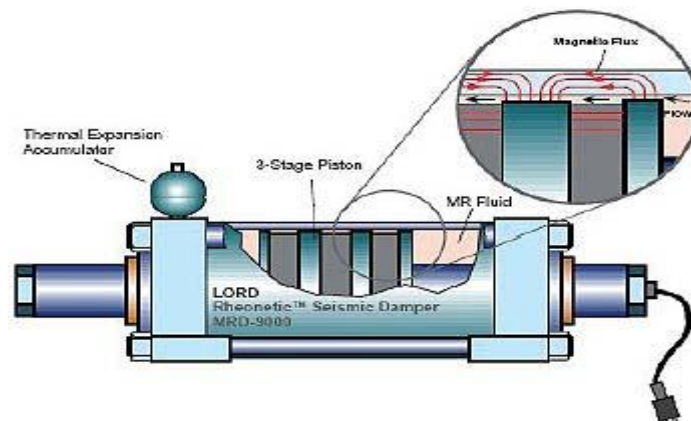


Figura 8: Amortiguador Magneto-reológico

### 2.4.2. FUNCIONAMIENTO DE UN AMORTIGUADOR MAGNETOREOLOGICO.

Se trata de un dispositivo que sirve para eliminar los efectos de oscilación producidos por un muelle. Al pasar por un bache, los resortes almacenan una cierta cantidad de energía, que, posteriormente, restituyen a través de balanceos. Los amortiguadores evitan que la carrocería del vehículo oscile continuamente (controlan los vaivenes transformando en calor la energía que acumula el muelle), disminuyen las variaciones de la carga dinámica de las ruedas y evitan que éstas salten sobre el suelo, proporcionando comodidad y seguridad.

Su principio de funcionamiento es sencillo; un pistón, unido a la carrocería a través de un vástago de fijación, se desliza en el interior de un cilindro ligado a la rueda y lleno de un fluido (aceite o gas). Una serie de orificios calibrados en el pistón permiten el paso del aceite entre las dos partes en las que queda dividido el cilindro, frenando así la oscilación del resorte en función de la resistencia que ofrece el fluido al pasar por los orificios.

En los amortiguador magneto-reológicos el fluido presente en el interior de estos está compuesto de aceite y micro esferas de hierro en suspensión, por lo que tiene propiedades magneto-reológicas, es decir, cambia sus propiedades físicas cuando se le somete a un campo magnético. En el interior se incorpora un pistón con una bobina y el campo magnético creado por ésta hace cambiar la resistencia a fluir del propio aceite entre los orificios del pistón, modificando su tarado (dureza) en milésimas de segundo, dependiendo de la corriente eléctrica que alimenta a la bobina. [armandotripoides.com / amortiguadores]

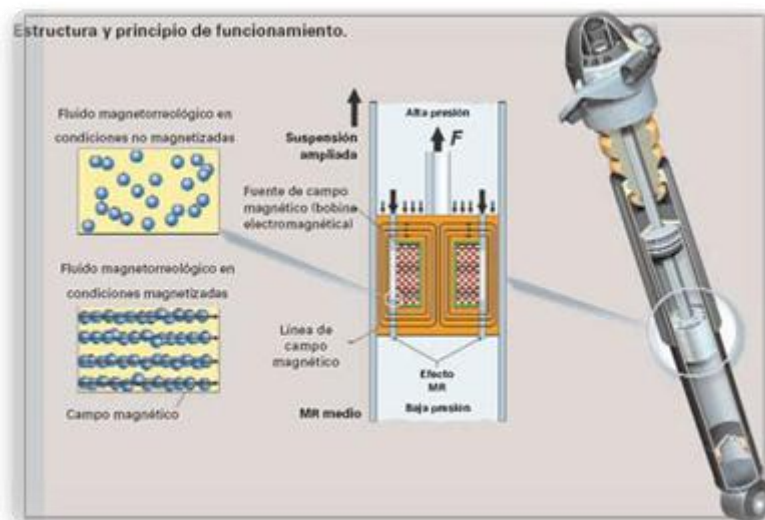


Figura 9: Funcionamiento Amortiguador



### 2.4.3. ELECCION DEL AMORTIGUADOR MR.

Son ya muchos los fabricantes que diseñan este tipo de tecnologías y ya muchas marcas de la industria automovilística que incorpora este tipo de tecnología a sus sistemas de suspensión.

El objeto de este proyecto es el estudio de control y diseño de este tipo de nuevas tecnologías aplicadas a seguridad activa en automóviles, aumentando su seguridad y respuesta durante su marcha. Se eligió este amortiguador debido a que es un modelo innovador de Lord que está teniendo gran éxito en el mercado.

De este modo conseguiremos un estudio de los nuevos mercados del sector de los sistemas de suspensión.

### 2.5. SOFTWARE DE DESARROLLO

Para el desarrollo y diseño de este proyecto se han usado principalmente dos herramientas o software de diseño:

- Orcad: es una herramienta CAD orientada a la simulación y diseño de elementos electrónicos.
- Keil U Vision: programa diseñado para la compilación, programación y depuración en C del microcontrolador ARM7.

### 2.5.1. DISEÑO ELECTRONICO: ORCAD

Como ya hemos reseñado anteriormente, ORCAD es una herramienta CAD muy completa y de lo más potente que actualmente existe en el mercado en lo referente a la simulación y diseño de componentes electrónicos.



Figura 10: Logo ORCAD

Orcad se concibe como un paquete de herramientas de diseño, nosotros principalmente utilizaremos dos:

-**Pspice**: para el diseño esquemático del sistema de control y para su simulación y estudio mediante graficas.

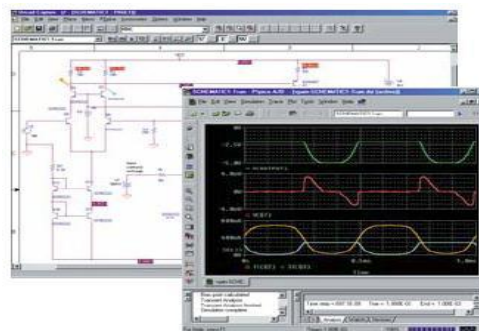


Figura 11: Pspice

-**Layout**: herramienta que sirve para una vez terminado nuestro diseño electrónico en Pspice y generado los archivos necesarios, permite diseñar nuestro sistema electrónico en una tarjeta PCB para posteriormente serigrafiarla en una placa especialmente fabricada para ello e implementar los componentes en ella.

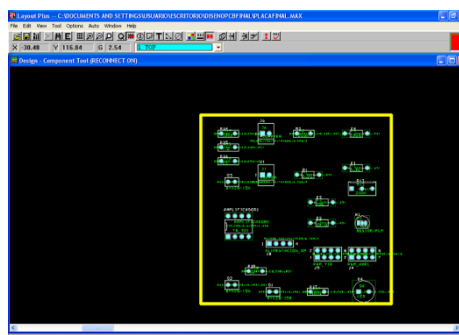


Figura 12: Layout

Se ha elegido este software por la potencia y sencillez en el uso de la misma, y por su versatilidad a la hora de introducir librerías de nuevos componentes y su simulación en el mismo

## 2.5.2. PROGRAMACION: KEIL $\mu$ VISION

Este software desarrollado por KEIL para el desarrollo, simulación y depurado de programas para los microcontroladores ARM nos ha permitido diseñar nuestra lógica de control que queremos ejercer sobre el amortiguador magneto-reológico.

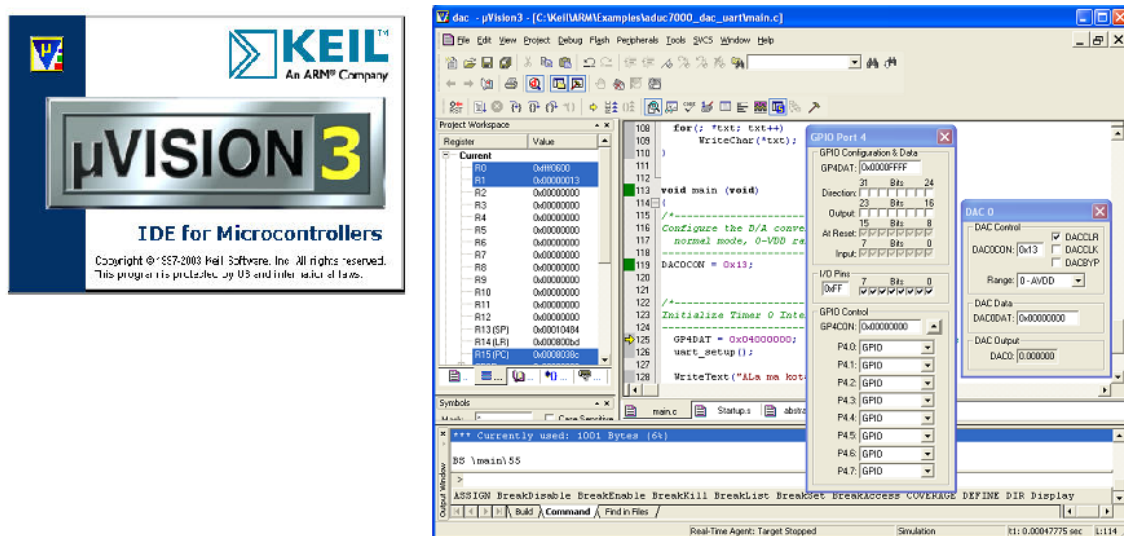


Figura 13: Keil U Vision 3

Nuestro programa diseñado consta de cinco partes

- Configuración del ARM:** Es el inicio del programa, en el cual decimos los ciclos de reloj, configuración de los puertos de entrada, salida, timer, etc.
- Configuración de los vectores de interrupción:** decimos en que parte de la memoria se ubicara la interrupción, duración de la misma y disparos de flanco.
- Bucle de trabajo:** es la parte esencial del programa, en el cual se trabaja constantemente con las señales de entrada para elaborar las respuestas a la salida.
- PID:** está incluido en el bucle de trabajo y va transformando la señal elaborada de salida a una respuesta que nosotros queramos obtener (más rápida o más lenta).
- Adquisición de valores mediante el ADC:** es la toma de niveles de tensión en analógica y su transformación en digital. También está incluido en el bucle principal y necesita un tiempo de operación entre toma y toma de niveles que debemos tener en cuenta si queremos que nos dé un resultado correcto.

## 3. DISEÑO

### 3.1. INTRODUCCION

Como anteriormente ya se ha mencionado, el desarrollo de este proyecto se puede describir en dos grandes apartados, el primero es el diseño electrónico del sistema, y el segundo es el diseño lógico del sistema.

El sistema electrónico lleva como objeto la adaptación o adecuación de las señales generadas por el ARM a el sistema del amortiguador magneto-reológico para el control de este y la adecuación de las señales recibidas del amortiguador hacia el ARM para que este las pueda interpretar correctamente, podemos definir nuestro sistema electrónico que hace las funciones de “transductor bilateral” o “transductor bidireccional”.

El programa o sistema lógico lleva como objeto portar la “inteligencia” o el sistema de control capaz de gobernar nuestro amortiguador para que actúe bajo las restricciones que nosotros le pongamos.

### 3.2. PARTE ELECTRONICA

La parte de electrónica la podemos dividir en tres partes diferentes:

- El microcontrolador es el encargado de gestionar las señales de entradas y dar la respuesta adecuada a esta mediante un tren de pulsos PWM para que el amortiguador actúe de una manera gobernable
- El circuito electrónico de control sobre el amortiguador magneto-reológico que proporciona una perfecta adecuación de señales en los dos sentidos para que exista un proceso preciso y “milimétrico” de nuestro sistema.
- El amortiguador magneto-reológico es el elemento final que adecuara las señales eléctricas recibidas por el microcontrolador para transformarlas en señales físicas y actuar de manera eficaz en el sistema de suspensión.



### 3.2.1. EL MICROCONTROLADOR

Para usar nuestro microcontrolador como parte de la aplicación desarrollada en este proyecto deberemos tener algunos conocimientos de uso, configuración y programación de este.

En la figura de abajo podemos observar cómo se conecta la placa al PC.

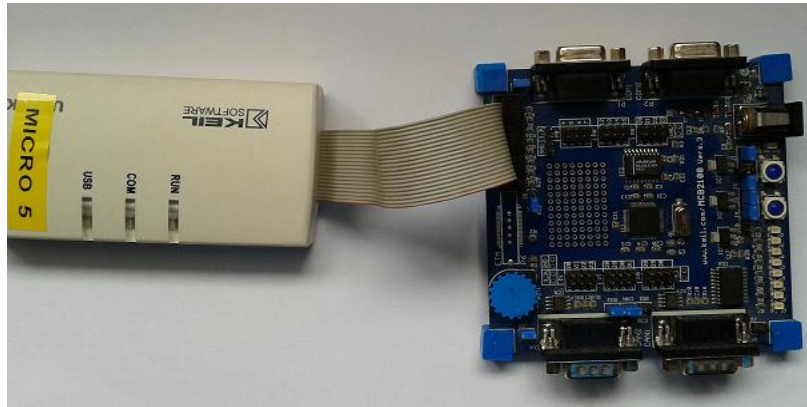


Figura 14: MCB2100 y Ulink

La conexión es simple, conectamos el depurador Ulink (creado especialmente para operar en la placa MCB2100) mediante USB al PC por un extremo, y por el otro extremo conectamos el depurador al anclaje JTAG.

Una vez hechas las conexiones, lo único que tendríamos que hacer es conectar la placa MCB2100 a la alimentación mediante un transformador de luz de 6 VDC, y ya estaría disponible para volcarle el programa de control y trabajar con él.

Lo segundo que deberíamos de hacer es conectar nuestra placa de control diseñada al microcontrolador. La conexión de la placa al MCB2100 es mediante los puertos y la conexión a 3,3 VDC y la propia tierra de la placa.

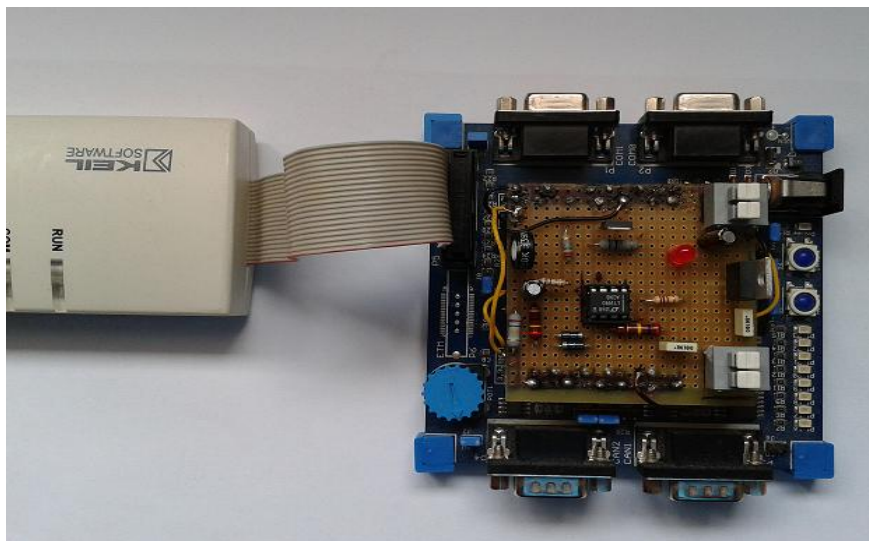


Figura 15: MCB2100 con Placa Electrónica

## Proyecto Fin de Carrera

Y posteriormente conectar nuestra placa al amortiguador magneto-reológico y a una alimentación de 12VDC para que alimente al amortiguador.

Para depurar nuestro programa, lo primero antes de conectar el PC directamente a todo nuestro sistema (MCB2100, placa control, amortiguador) es simular nuestro proceso mediante un simulador de señales que dispone el software de keil para ver si realmente está bien diseñado nuestro programa.

Para crear nuestro programa para la placa deberemos seguir unos sencillos pasos:

- Primero, crearemos un proyecto, en Project>New
- Segundo, en ese proyecto incluiremos el startup correspondiente al LPC2129
- Tercero, incluiremos las hojas donde se encuentra nuestro código en el proyecto junto al startup.

Una vez creado y diseñado nuestro proyecto entero, le daremos a built target para ver si contiene errores y los depuraremos hasta que no contenga ninguno. Cuando conseguimos que nuestro programa este limpio de errores de programación, procederemos de nuevo a dar a built target, y después en la barra de herramientas en Debug> Start, Stop

Estos pasos son los que debemos hacer cada vez que queramos compilar nuestro código o mejor expresado, nuestro proyecto diseñado, pero el programa Keil u Vision, como hemos dicho anteriormente ofrece dos modos de comprobación y depuración del código, uno es un simulador de señales y el segundo es hacer funcionar el código sobre la misma MCB2100.

Para usar el modo de simulación de señales tendremos que hacer lo siguiente:

Primero pincharemos en la rama principal del proyecto y hacemos click en el botón derecho y le damos a “options for target”

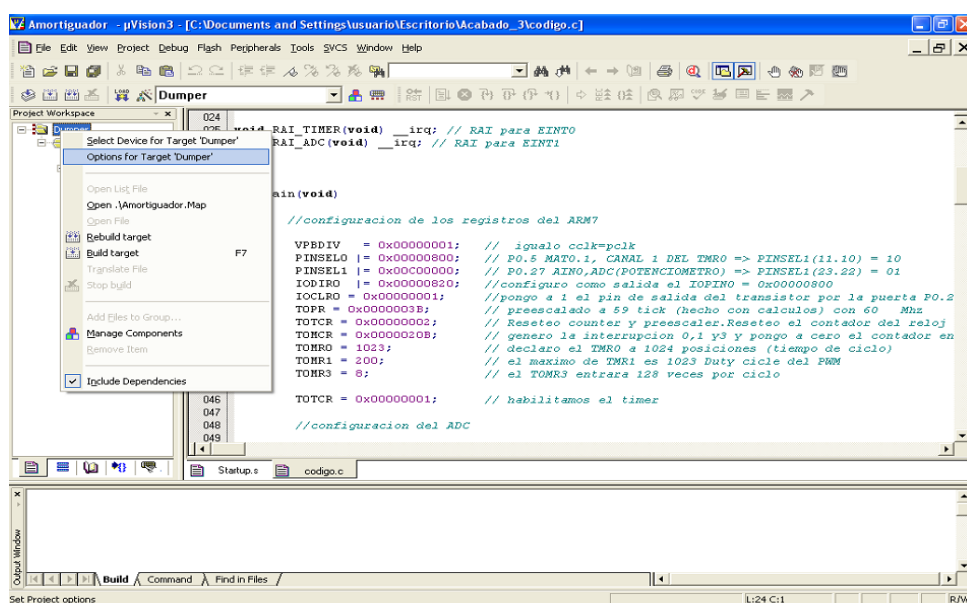


Figura 16: Configuración debugger 1

Después en la opción de Debug, deberemos señalar la opción de “use simulator”

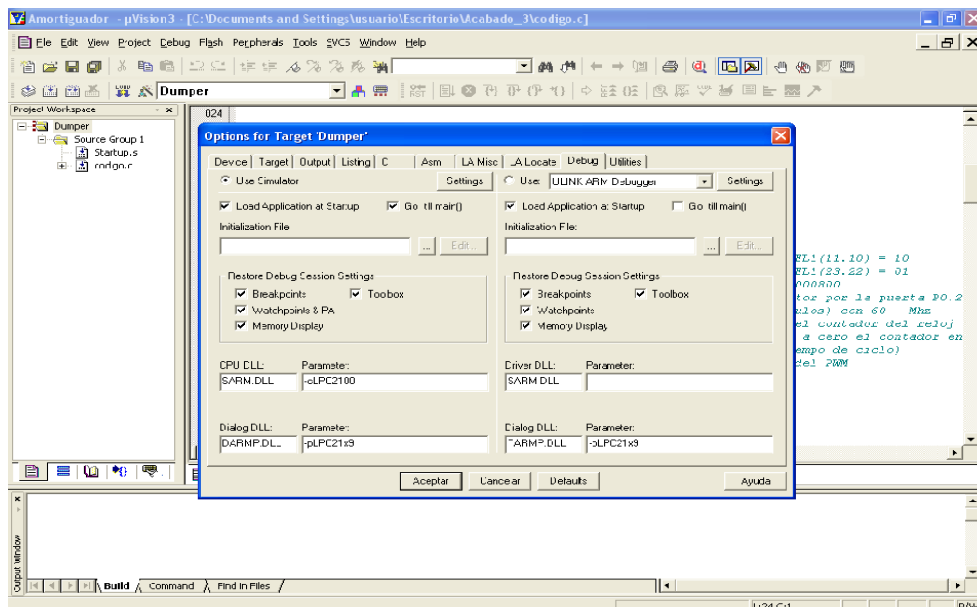


Figura 17: Configuración debugger 2

Después y por ultimo en utilities elegimos la opción de “Use External toolbox Flash”

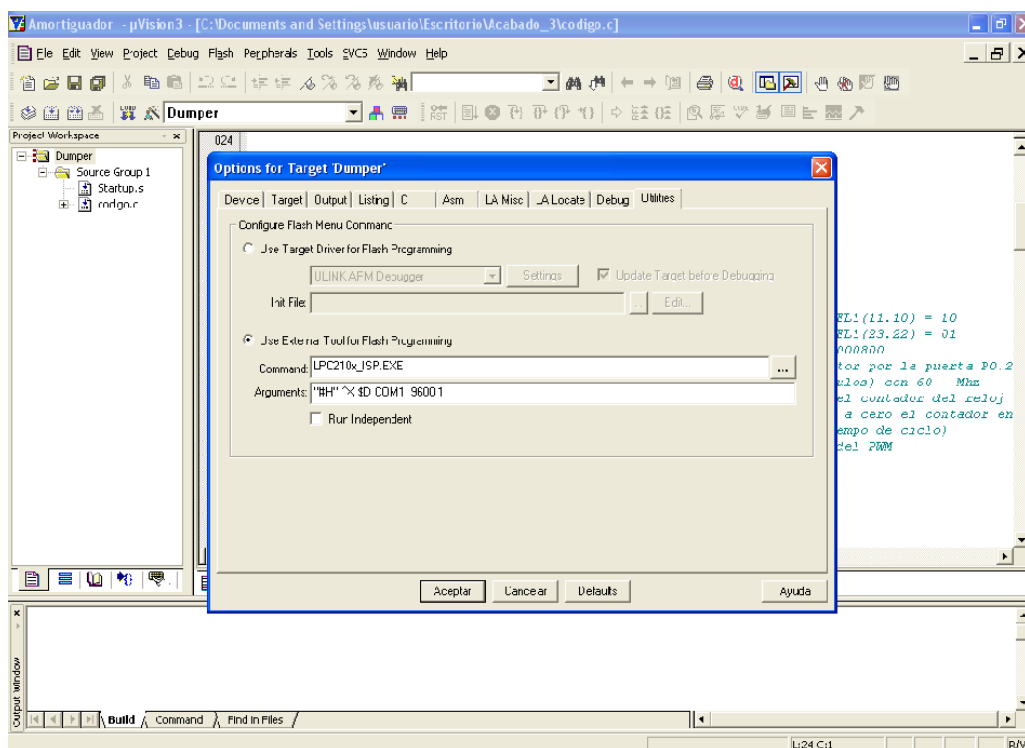


Figura 18: Configuración debugger 3

Y ya, lo único que nos quedaría es dar a debug> Start y comenzar a simular con la señales que queramos probar en nuestro simulador del programa.

El entorno de simulación del programa sería el siguiente

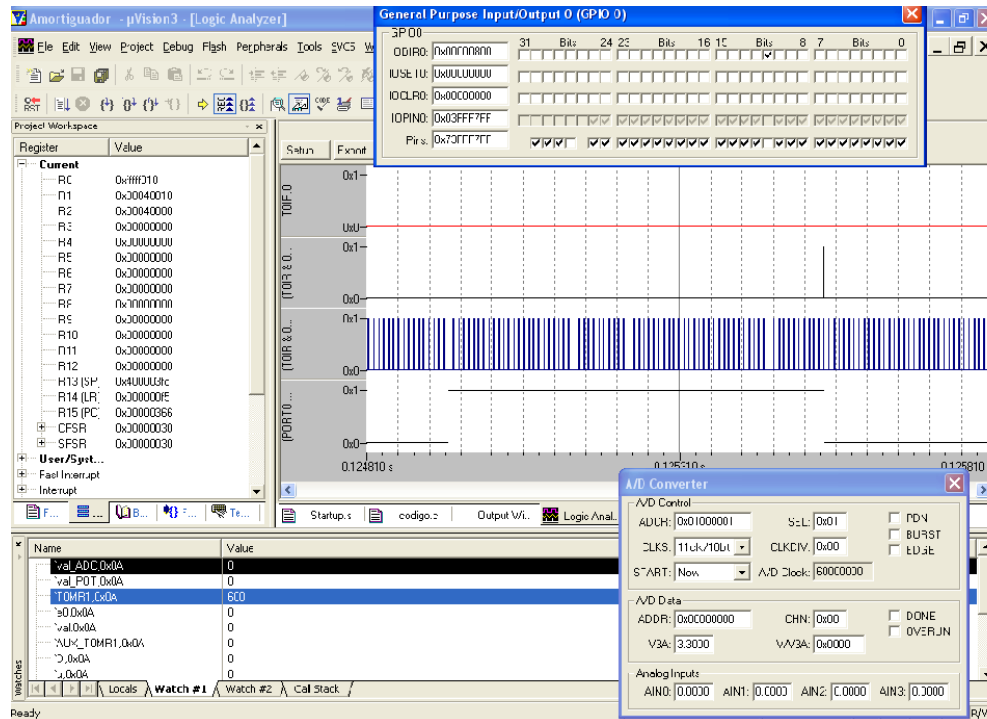


Figura 19: Simulador de Keil

Ahora si lo que queremos es probar el programa sobre la placa MCB2100 para comprobar que la simulación del programa deberemos variar los siguientes parámetros.

Lo que deberemos hacer es; de nuevo irnos a options target, (dando al botón derecho) y en las opciones de nuevo cambiamos en debug la opción de use simulator por la opción de “Use U link”

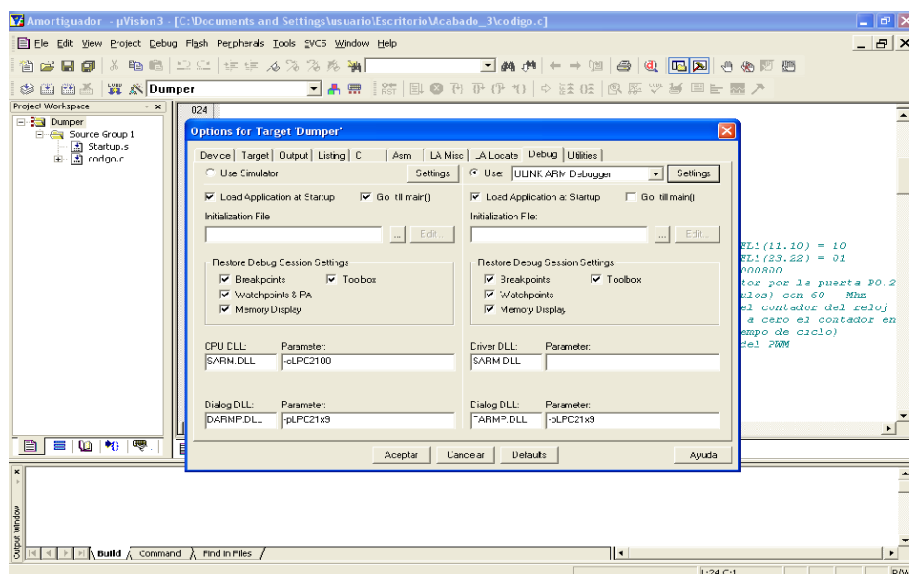


Figura 20: Configuración Debugger 4

Y por ultimo en la opción de utilities seleccionaremos la opción de “Use target” y marcaremos la casilla de “Update Target” para que el programa vuelque todo el código en el microcontrolador y se actualice, y de este modo nos aseguramos que el microcontrolador actúe con el último programa creado.

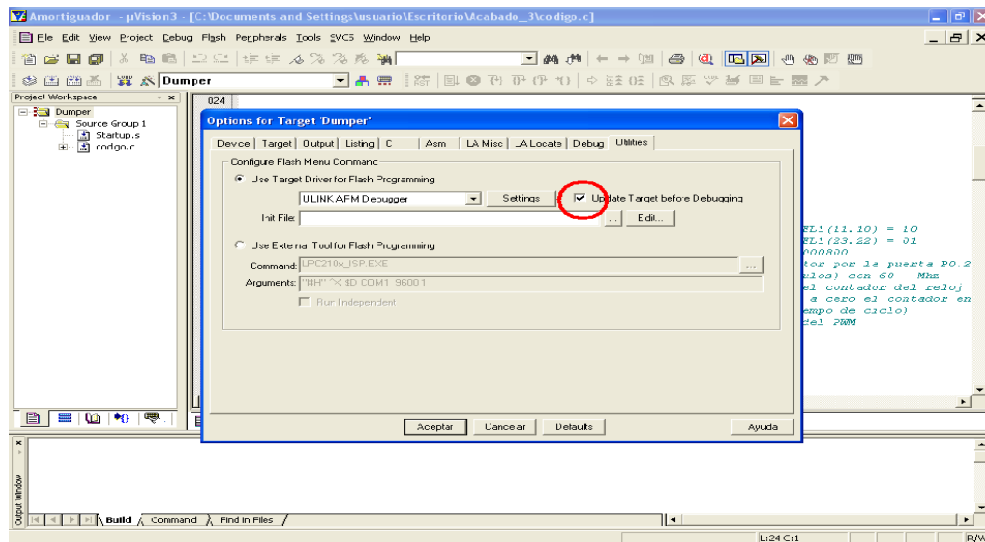


Figura 21: Configuración Debugger 5

Y lo siguiente que haremos será depurar nuestro código funcionando en tiempo real.

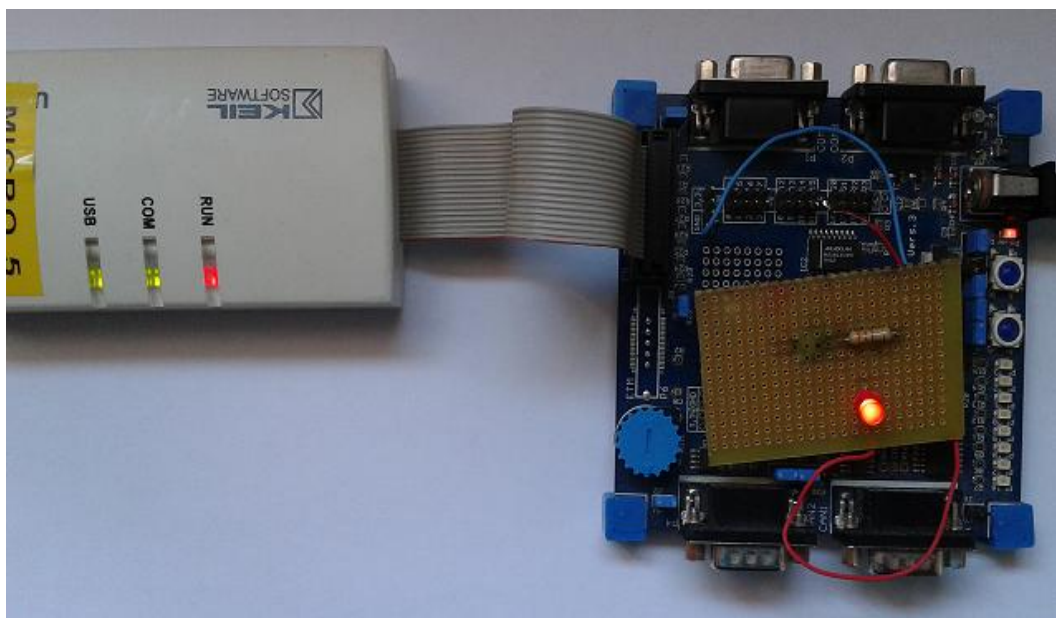


Figura 22: Depuración en tiempo real

## 3.2.2. CIRCUITO DE CONTROL

El circuito diseñado que nos permitirá poder gobernar nuestro amortiguador magneto-reológico es de simple aplicación, se basa en el ancho de pulso generador por la intensidad que llega del amortiguador y que modula el mosfet según la configuración que le demos, depende de este si trabaja en lazo abierto o lazo cerrado, si esta en continuo o va regulado con un PID.

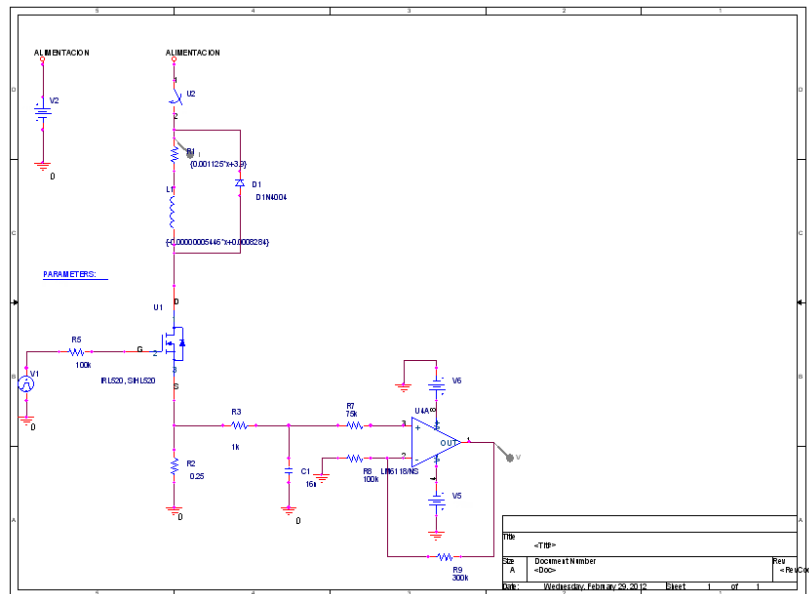


Figura 23: Esquemático de Potencia

Y la señal PWM a controlar será la siguiente

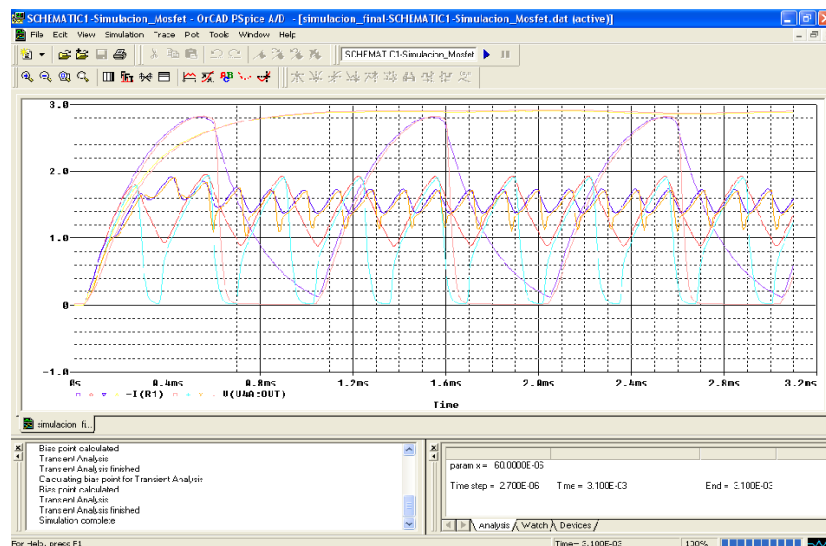


Figura 24: Simulación Señales

Es sobre esta señal que nos da el circuito de control, sobre la cual deberemos estudiar nuestro sistema completo y como tratarlo (filtrado, frecuencia etc.)



## 3.2.3. AMORTIGUADOR MR (CARACTERIZACION)

Realmente cuando se afronto el reto de conseguir desarrollar todo este proyecto como se pedía, lo primero que se intento estudiar fue, como caracterizar el amortiguador como un componente eléctrico o electrónico.

Lo primero, y para hacernos una idea concreta, lo que hicimos fue mirar las hojas de características del amortiguador, entre las cuales se extrajo los siguientes valores

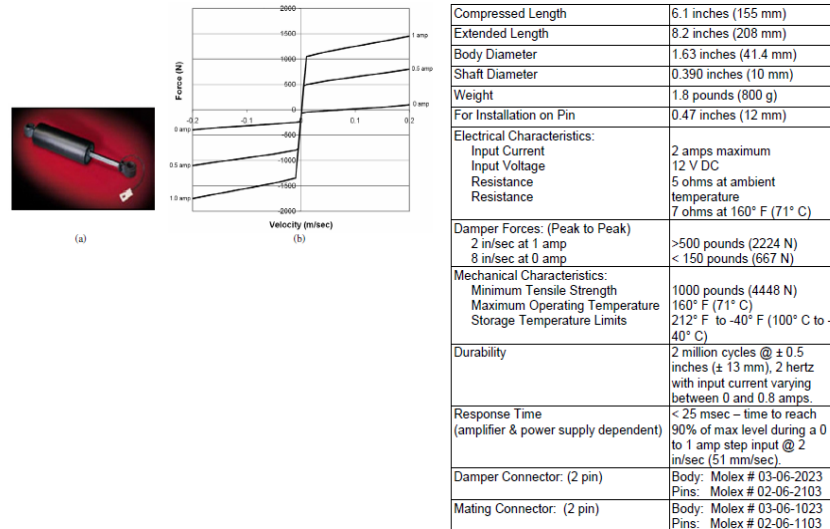


Figura 25: Hoja características amortiguador

En las cuales se puede observar que el amortiguador se comporta aparentemente como un componente electrónico pasivo.

También se extrae de las hojas de especificaciones las respuestas que se obtiene del amortiguador ante diferentes ensayos

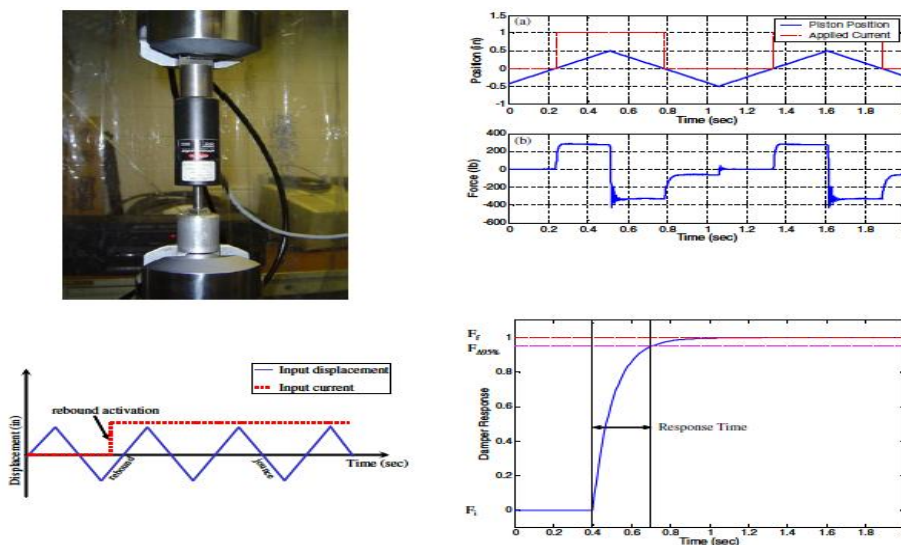


Figura 26: Ensayos amortiguador

## Proyecto Fin de Carrera

Una vez que tenemos una idea sobre el funcionamiento de este sistema de amortiguación, decidimos quedarnos con las siguientes graficas como datos y trabajar sobre ellas.

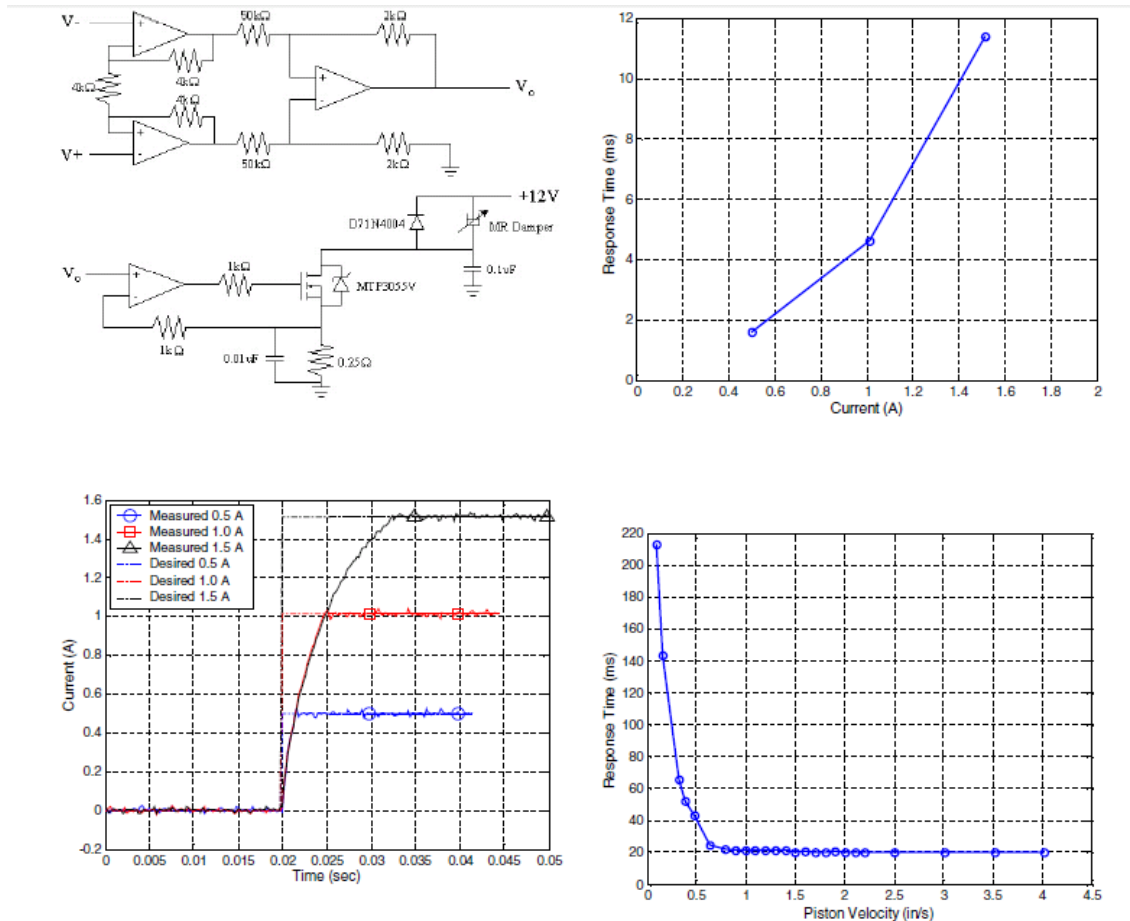


Figura 27: Graficas Amortiguador

Lo interesante fue ver el sistema de control que proponían para el amortiguador, y la respuesta que obtenía frente al tiempo al pasar corriente por el amortiguador. De ahí empezaron todas las deducciones.

Del circuito electrónico que va conexasionado al amortiguador MR, y de las curvas características “Amperios-Tiempo” se deberá sacar el modelo eléctrico del amortiguador MR.

Lo que haremos ahora será, estudiar el comportamiento de los elementos en los circuitos y ver sus respuestas:



Primero vamos a analizar la bobina. Si tenemos un circuito puro L, cuando este le sometemos a una fuente DC, lo que obtenemos como señal de intensidad en la bobina es la siguiente:

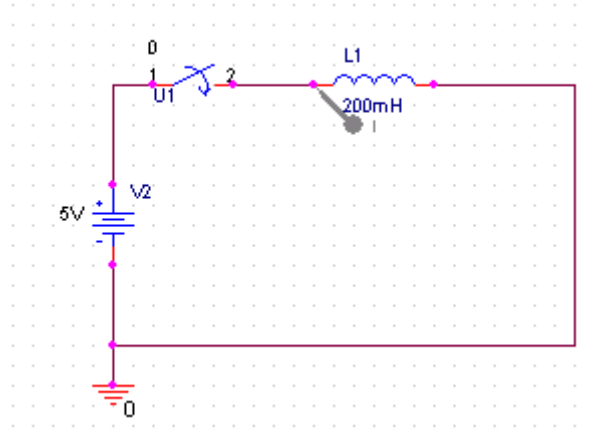


Figura 28: Circuito L

Vemos que ante una señal continua de tensión obtenemos una recta de corriente en la bobina

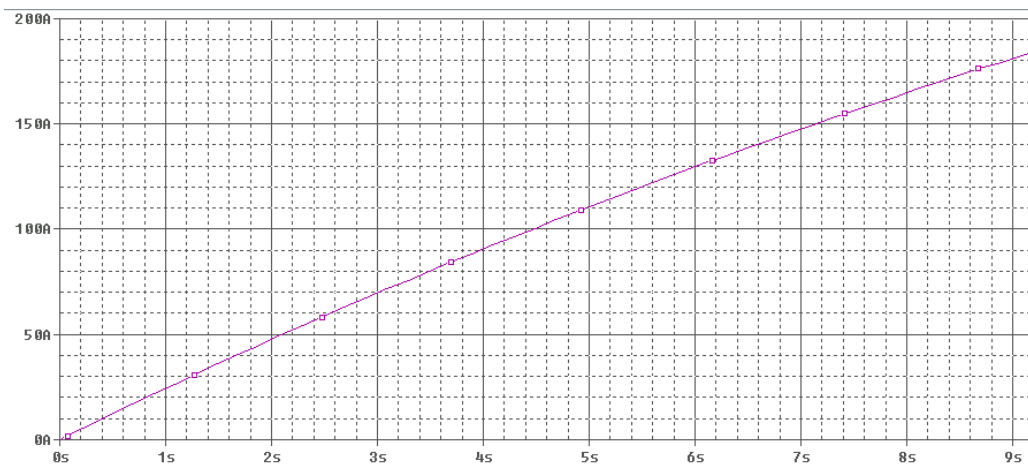


Figura 29: Grafica Circuito L

Y si ponemos una generación de onda cuadrada de tensión:

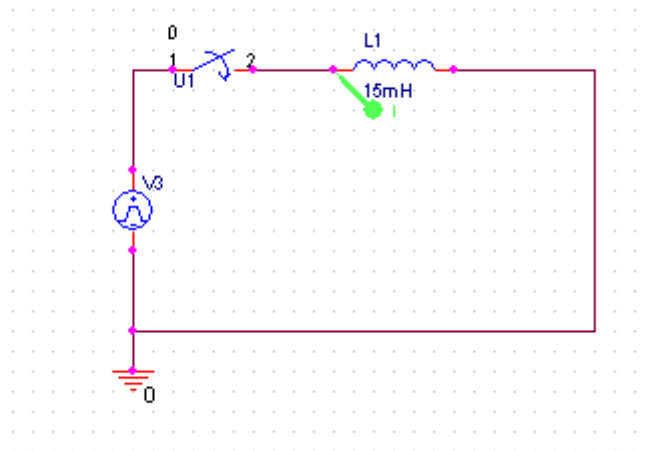


Figura 30: Circuito L

La señal de intensidad en la bobina será la siguiente:

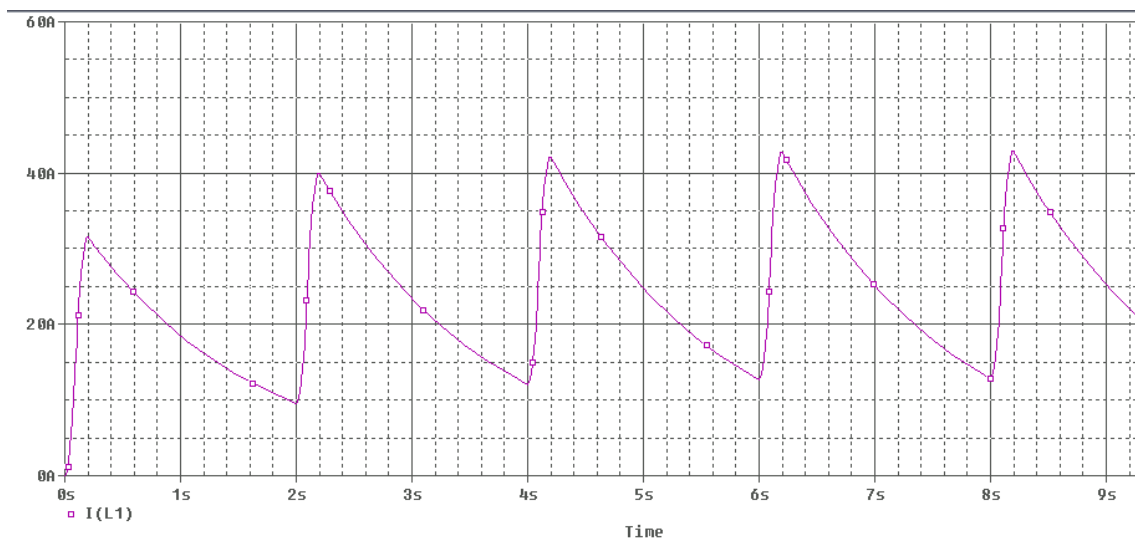


Figura 31: Grafica circuito L

Es lógico, debido a que son las rectas de carga y descarga de la bobina.

Y si ahora tenemos un circuito RL:

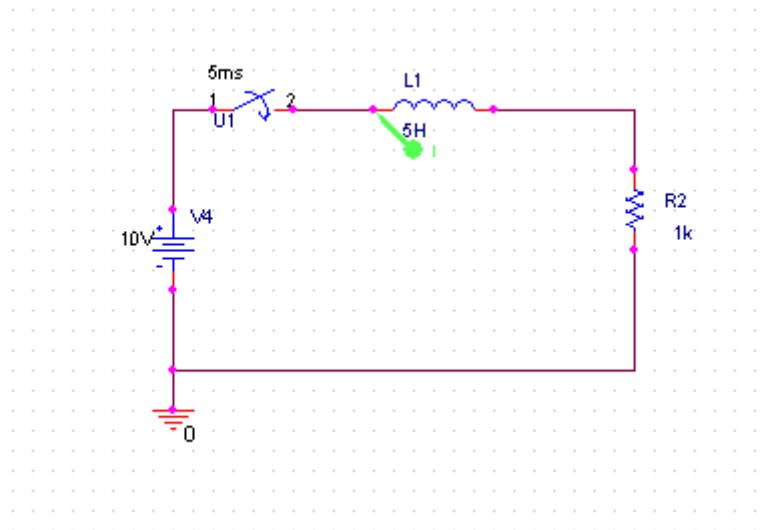


Figura 32: Circuito RL

La intensidad por la bobina será:

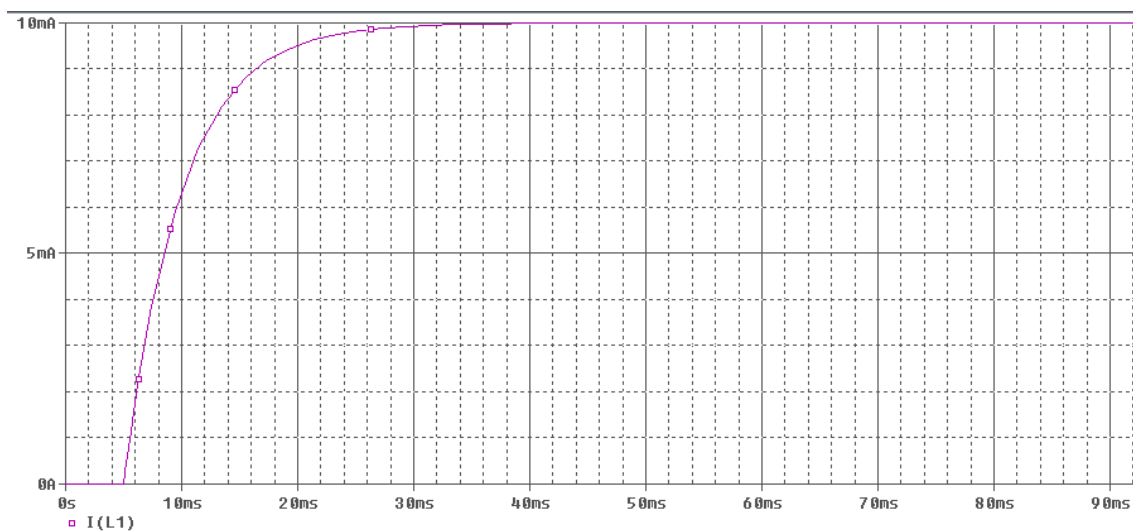


Figura 33: Grafica circuito RL

Y al estabilizarse en 10 mA esta seguirá constante mientras no se modifique nada en el circuito.

Si ahora tenemos un circuito RC:

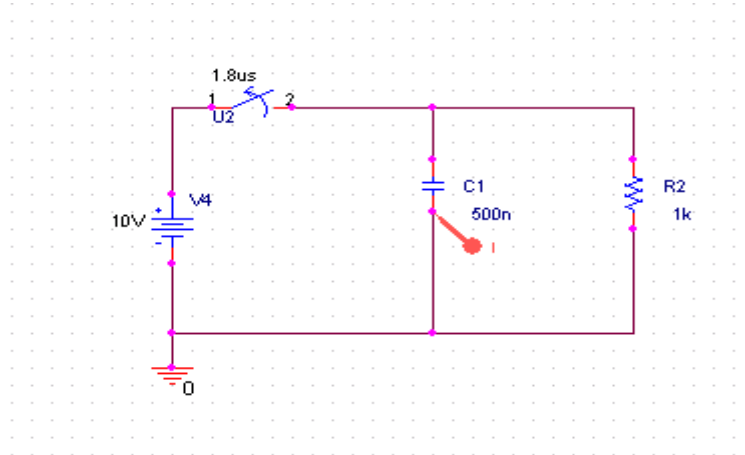


Figura 34: Circuito RC

Haremos dos simulaciones para observar las curvas de carga y descarga de este condensador.

Primero lo que haremos será, con el interruptor cerrado del circuito a los 1.8 us abriremos el circuito y observaremos como pasa de ser cero la intensidad (debido a que el condensador está cargado a tope), al abrirse el interruptor de la fuente de tensión, el que tiene que alimentar ahora la resistencia es el condensador solo, con lo que la veremos subir a su máximo de intensidad en una curva:

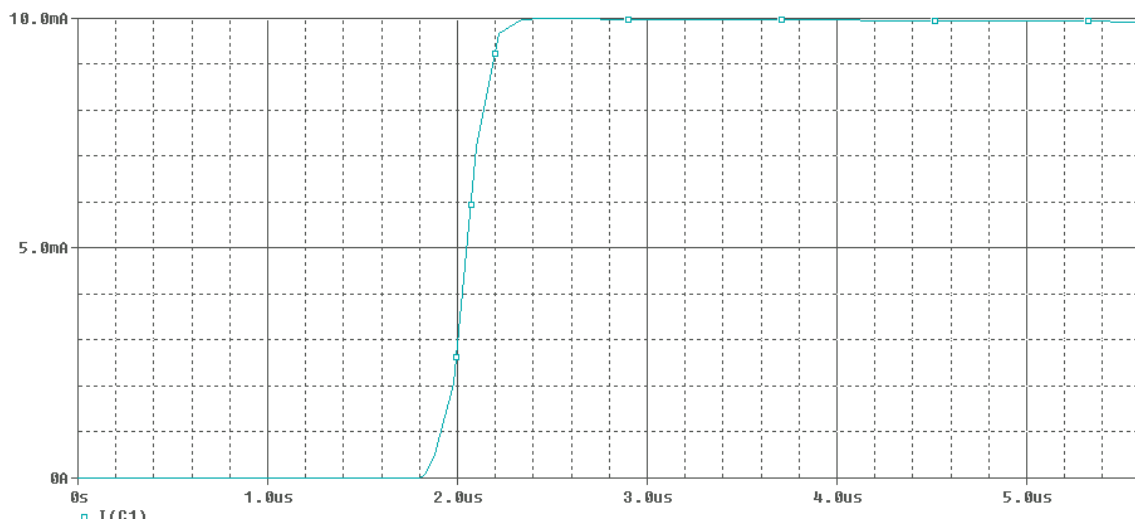


Figura 35: Grafica circuito RC

Pero si esta simulación le ampliamos el tiempo de funcionamiento, observamos que la intensidad vuelve a caer a cero más lentamente, debido a que toda la energía en el condensador se disipa:

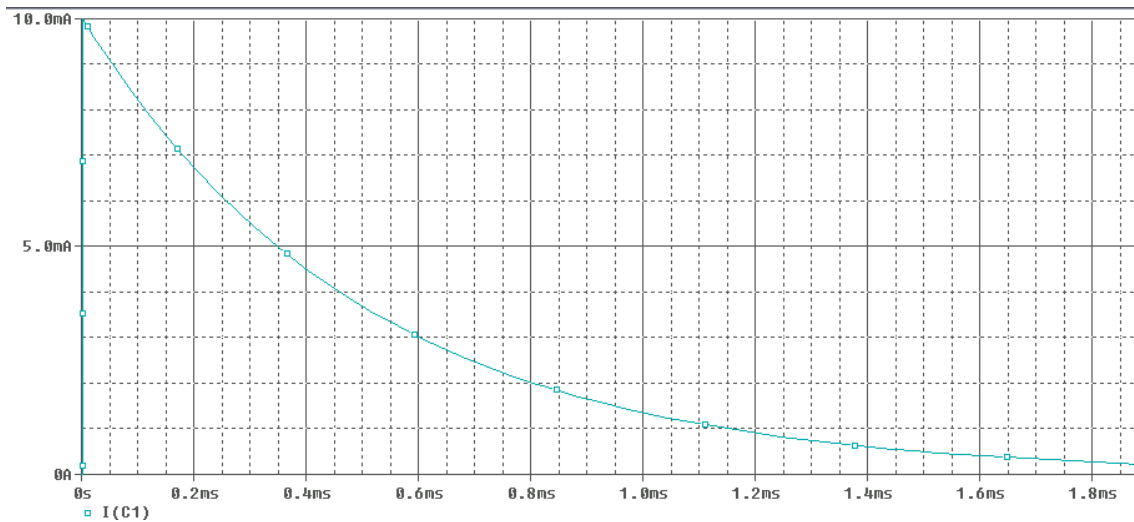


Figura 36: Grafica 2 circuito RC

El caso contrario ocurre cuando medimos la tensión del condensador, que vemos que está cargado a 10 V, y en cuanto abro el interruptor en el instante 1.8 us, la tensión empieza a disminuir:

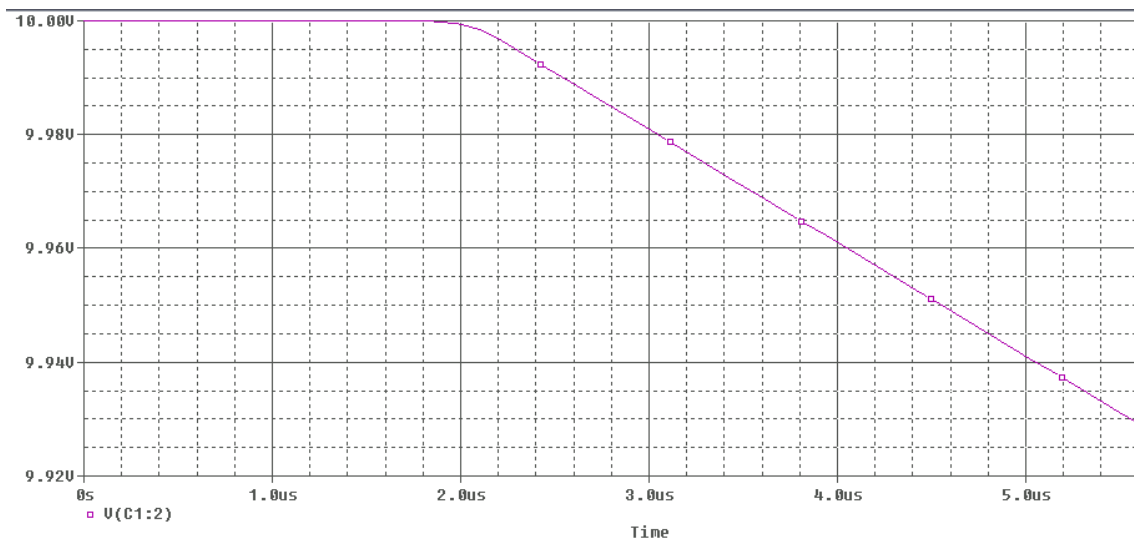


Figura 37: Grafica 3 circuito RC

Y si ampliamos más el rango de tiempo de simulación lo veremos más claramente:

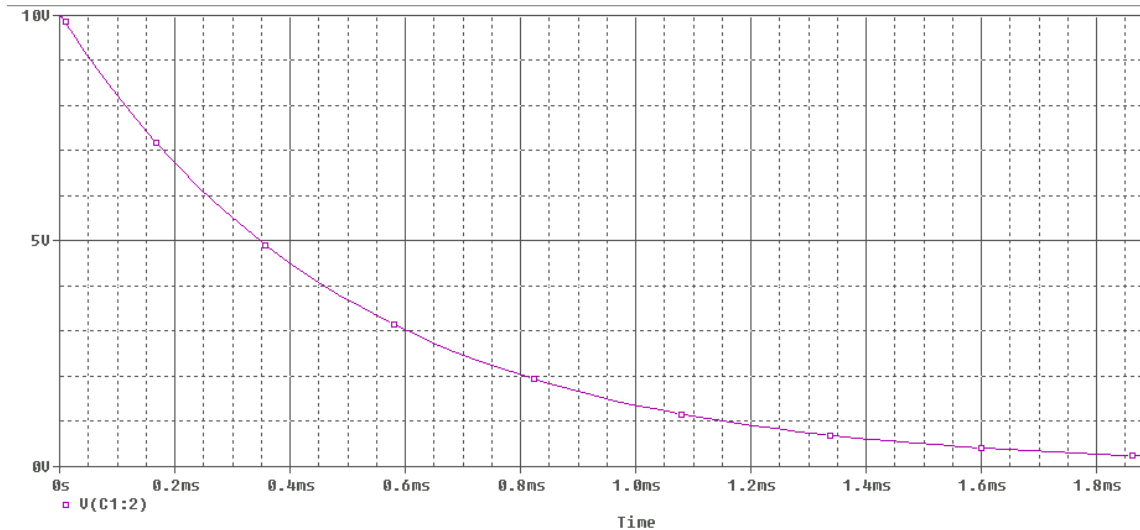


Figura 38: Grafica 4 circuito RC

Ahora haremos el segundo caso, es decir mediremos la recta de carga del condensador con la corriente, para ello abriremos de inicio el interruptor de la tensión y con el condensador vacío cerraremos el interruptor para ver la curva de intensidad de la carga:

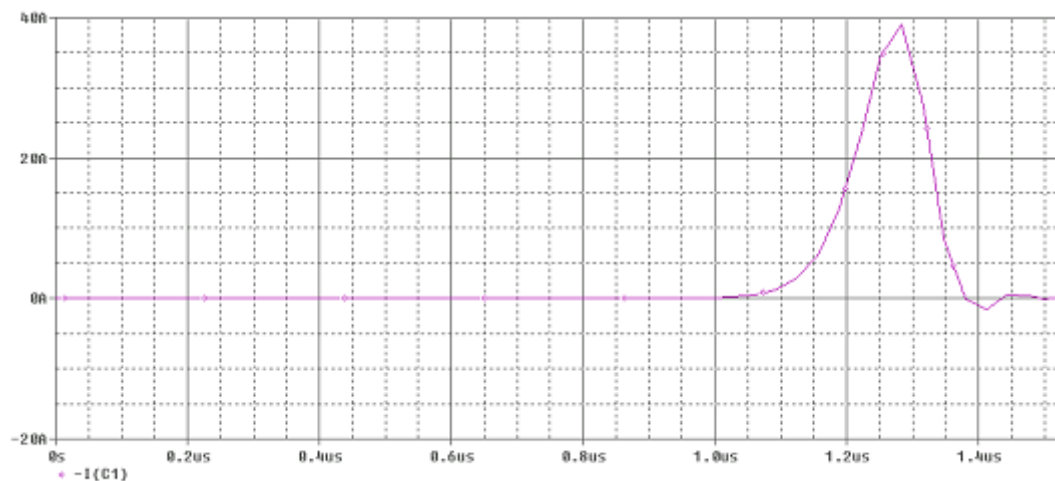


Figura 39: Grafica 5 circuito RC

Vemos que de nuevo va subiendo la intensidad, hasta cierto intervalo, en el cual comienza a decrecer, esto es debido a que en cuanto el condensador se va llenando de energía comienza circular menos corriente por él.

En cambio con la tensión sucede al contrario:

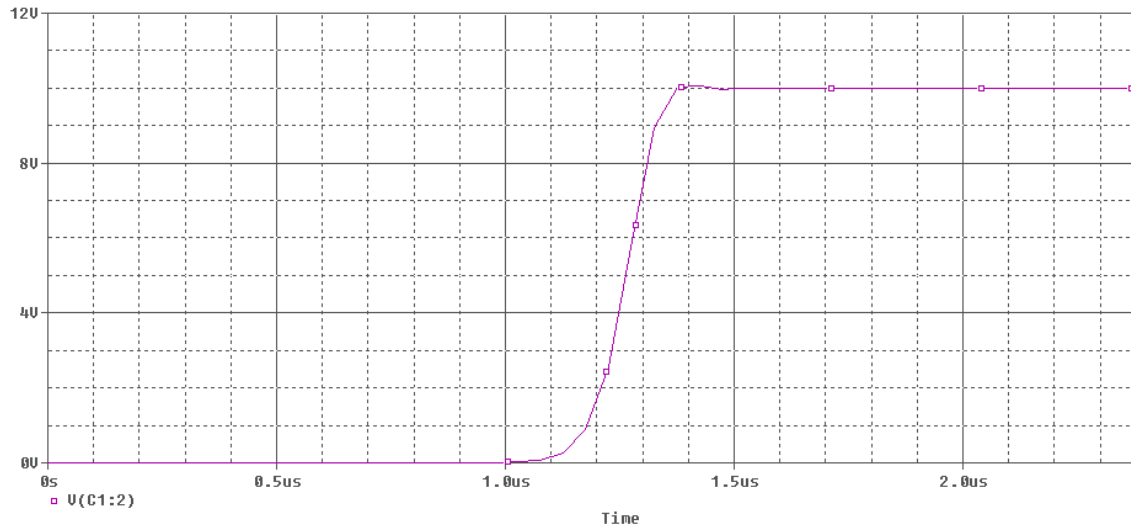


Figura 40: Grafica 6 circuito RC

Una vez llenado el condensador, la tensión en él se mantiene constante, mientras no se vacíe de energía de nuevo.

Visto el comportamiento de la bobina y el condensador así como el circuito RC y el circuito RL, lo que haremos será probar estos circuitos en nuestro esquema electrónico de tensión.

Sabemos que por la hoja de características, ante un impulso en el circuito obtenemos la siguiente grafica:

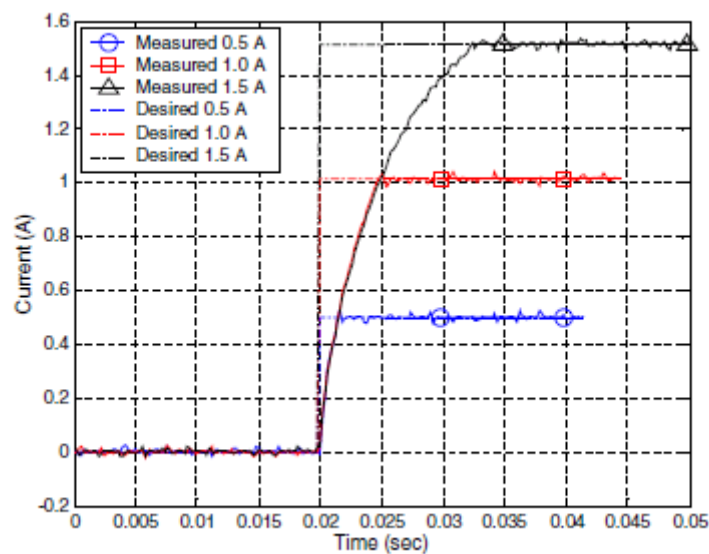


Figura 41: Respuesta amortiguador en Intensidad

Es decir, una logarítmica creciente, hasta que se satura y se mantiene constante.

Está claro, que para ver y conseguir la misma grafica, tendremos que hacer la misma prueba, es decir, en nuestro mosfet le meteremos por la base una tensión constante, para que deje pasar la intensidad hacia el colector, e instalaremos un interruptor al principio de la fuente de tensión, así este modelo nos será válido para obtener nuestra curva de corriente deseada que será objeto de estudio.

Ahora empezaremos a probar cual es nuestro modelo, modelando primero el amortiguador como un condensador:

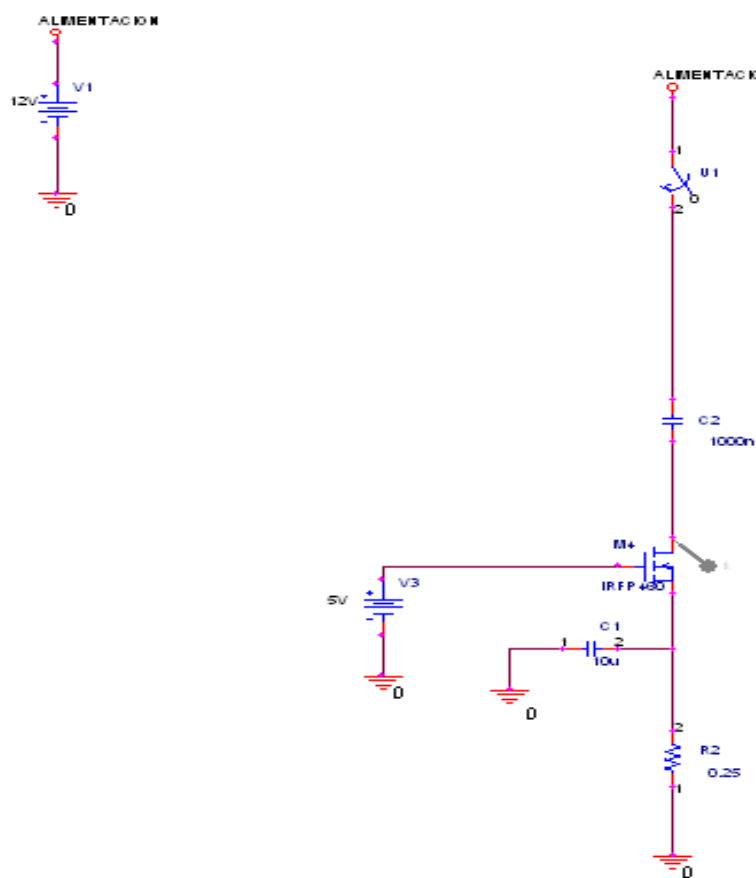


Figura 42: Circuito de Potencia



La grafica de la intensidad es:

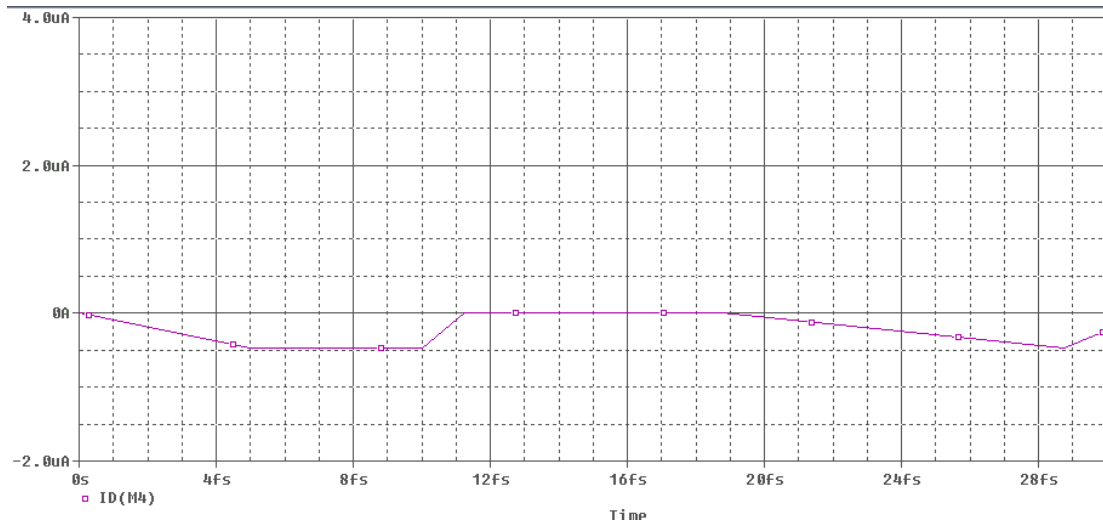


Figura 43: Grafica Circuito de Potencia

Vemos que ni los tiempos, ni el comportamiento se asemejan a la grafica “Amperios-Tiempo” que nosotros buscamos.

Si ahora modeláramos con un condensador y resistencia en serie obtenemos:

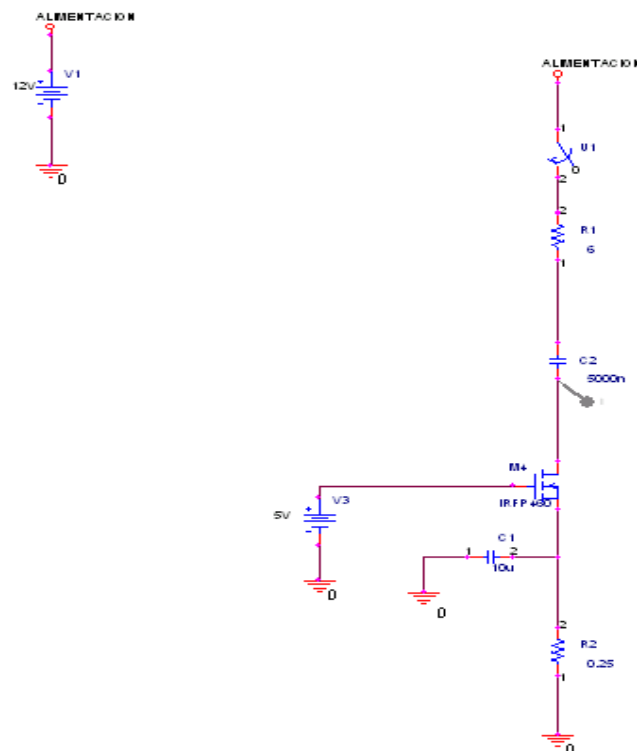


Figura 44: Circuito de potencia 2

La intensidad es:

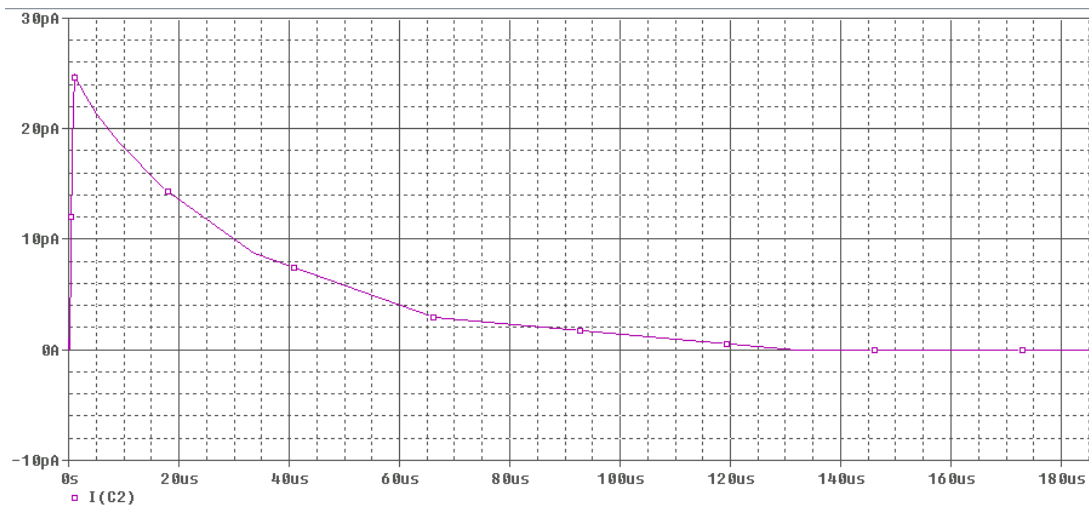


Figura 45: Grafica circuito de potencia 2

Vemos que es una curva la que se crea, pero esta tiende a cero, por lo que no nos es válida como solución.

Si hacemos la simulación en menos tiempo, obtenemos mejor el inicio de la subida de la curva:

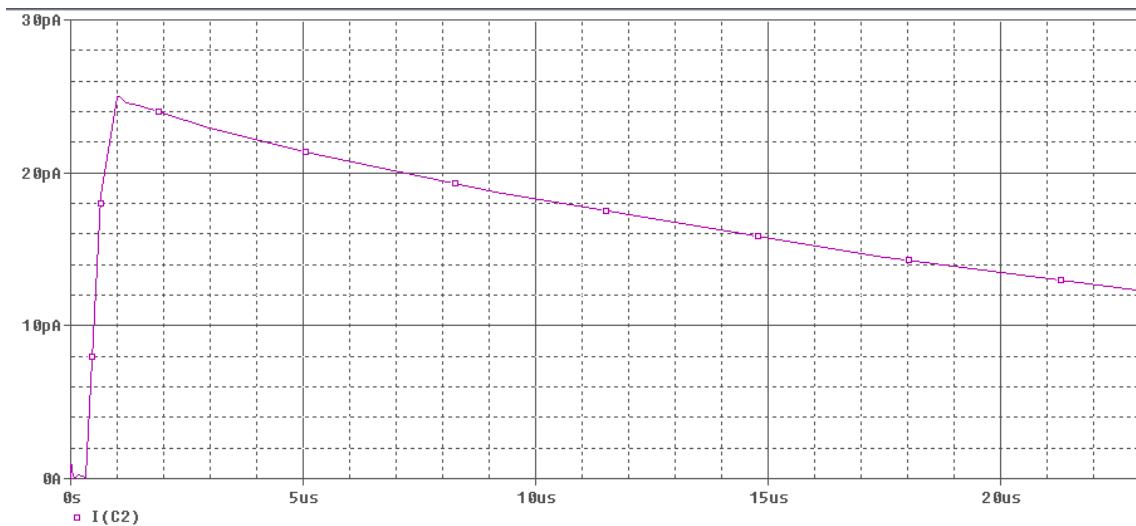


Figura 46: Grafica 2 circuito de potencia 2

Donde vemos que aunque presente una curva subida, esta no se mantiene y comienza otra curva de bajada que tiende a cero.

Ahora modelamos con un condensador en paralelo con una resistencia.

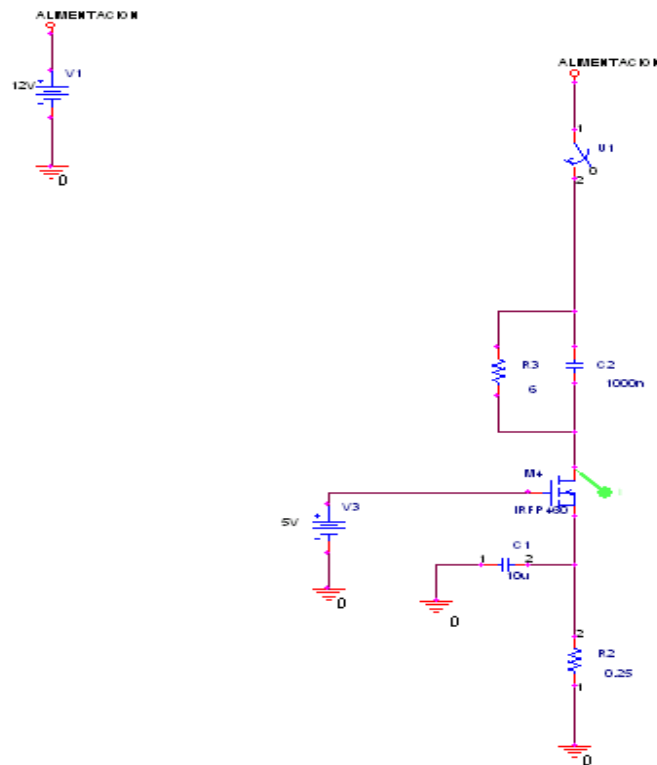


Figura 47: Circuito de Potencia 3

Y esta es la grafica de la intensidad:

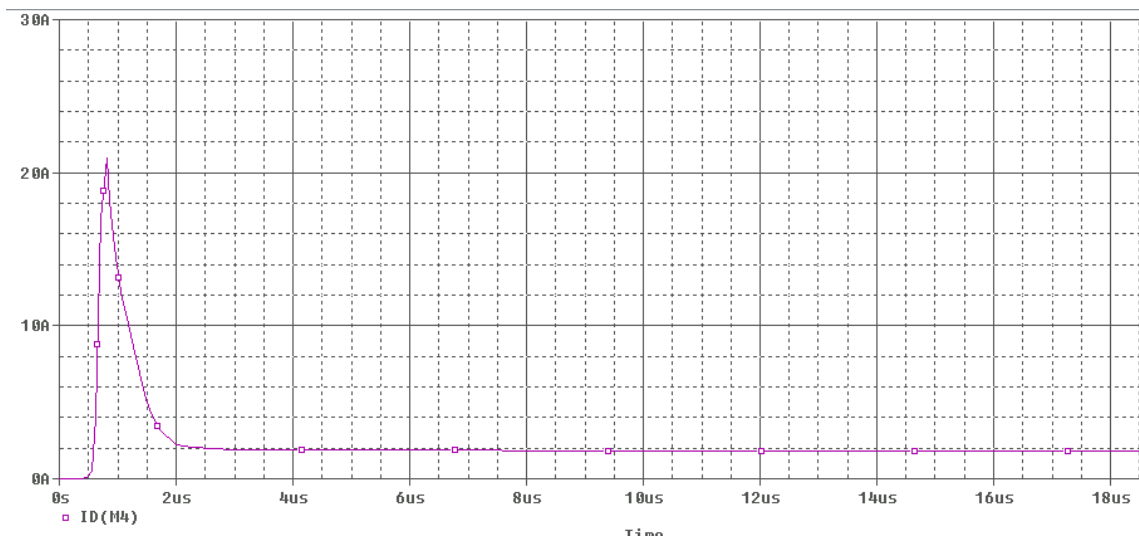


Figura 48: Grafica circuito de potencia 3

Vemos que tiene un comportamiento similar a la anterior y no nos sirve como respuesta.

### Typical Data\*

Compressed Length	6.1 inches (155 mm)
Extended Length	8.2 inches (208 mm)
Body Diameter	1.63 inches (41.4 mm)
Shaft Diameter	0.390 inches (10 mm)
Weight	1.8 pounds (800 g)
For Installation on Pin	0.47 inches (12 mm)
Electrical Characteristics:	
Input Current	2 amps maximum
Input Voltage	12 V DC
Resistance	5 ohms at ambient temperature
Resistance	7 ohms at 160° F (71° C)
Damper Forces: (Peak to Peak)	
2 in/sec at 1 amp	>500 pounds (2224 N)
8 in/sec at 0 amp	< 150 pounds (667 N)
Mechanical Characteristics:	
Minimum Tensile Strength	1000 pounds (4448 N)
Maximum Operating Temperature	160° F (71° C)
Storage Temperature Limits	212° F to -40° F (100° C to -40° C)

Página 51

Al hacer la simulación nos queda una grafica muy parecida en intensidad a la que buscábamos. De este modo nos queda ajustar los parámetros a 7.5 ohm y 20mH y poner al interruptor que cerrara a los 0,02 sg. Al hacer la simulación en estas condiciones nos da una grafica exacta a la de las especificaciones:

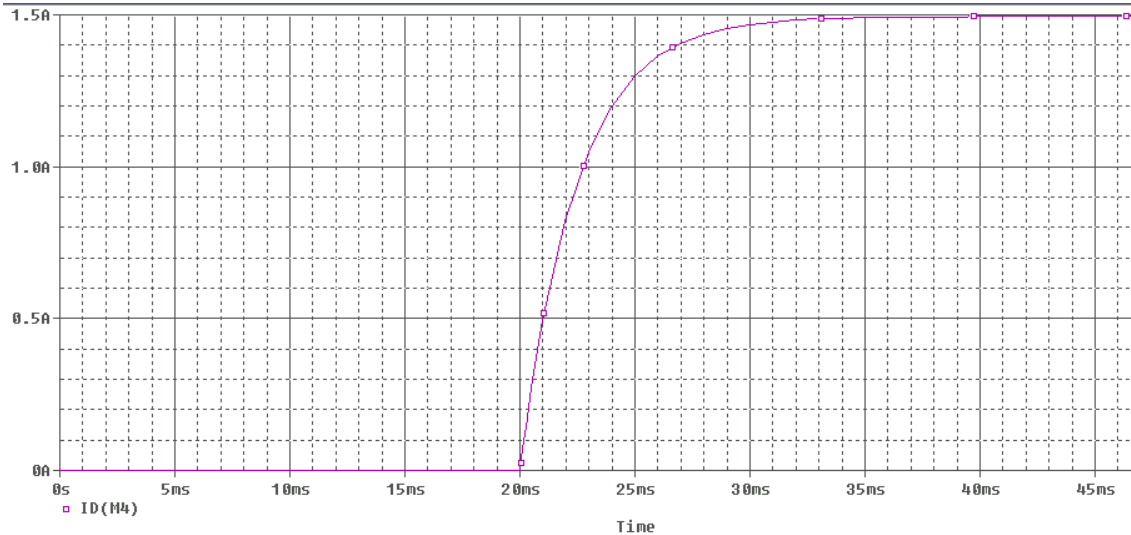


Figura 51: Grafica circuito de potencia 4

Comparando con la original:

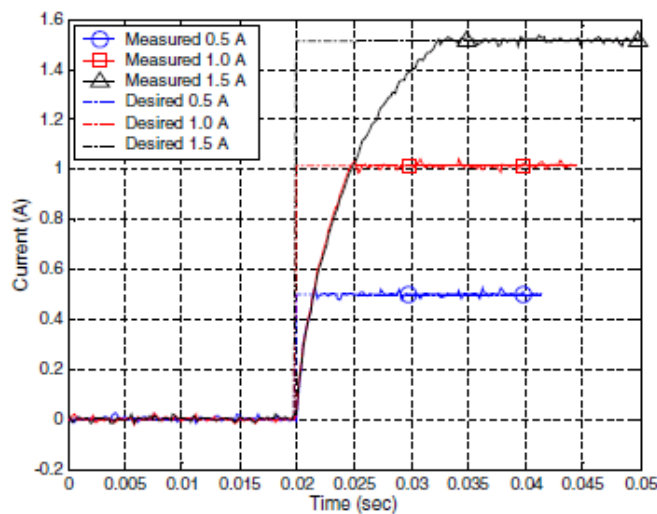


Figura 52: Valores Intensidad amortiguador

Veo que la grafica hallada es idéntica para 1,5 Amperios, que tiene una subida de 0.02 hasta 0.033 donde alcanza la saturación y luego se mantiene ahí fija.

Comparando las simulaciones con condensador, veo que siempre la corriente en estos se hace cero, es lógico, cuando un condensador se “llena” deja de circular corriente por él, hasta que le obliguemos a descargarse, entonces circulara de nuevo corriente de él hacia el resto del circuito.

## Proyecto Fin de Carrera

Además volviendo a nuestro esquema de control del amortiguador, resulta ilógico que la corriente se haga cero, debido a que si deja de circular corriente por el amortiguador, este dejara de perder las propiedades de fluidez que andamos buscando para él en ese instante. Por lo tanto, se nos hace sensato utilizar el circuito RL, debido a que una vez alcanzado el fin del transitorio, entrara en saturación y se mantendrá constante mientras no le variemos la tensión.

Otra pista que lleva a pensar que es un circuito RL es el diodo en anti-paralelo del circuito que me propone las hojas de características:

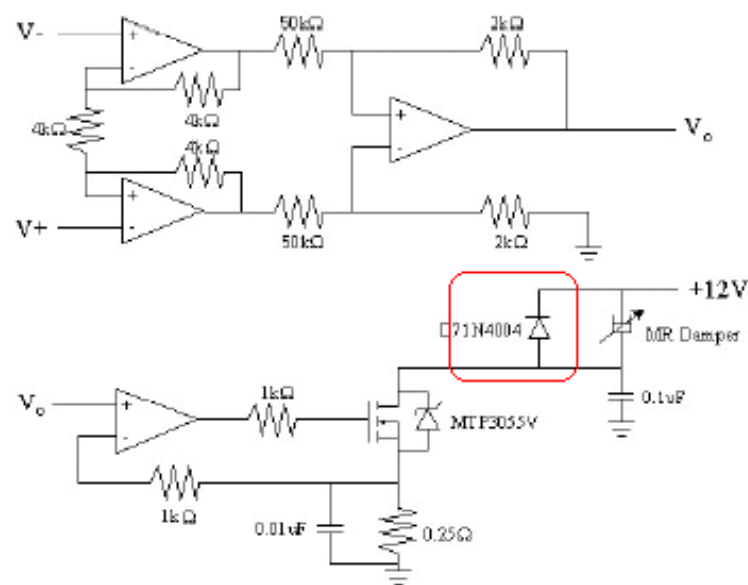


Figura 53: Electrónica control amortiguador

Si bien, estudiando conceptos de electrónica de potencia, circuitos y maquinas eléctricas, el diodo en anti-paralelo se ponía en paralelo con la bobina o circuito RL, para que la bobina disipara su inercia y no obligara a conducir en tiempos en los cuales no debía conducir.

Por ejemplo, ahora vamos a poner al modelo un generador de pulsos en la base (realmente es así como es nuestro modelo, un control por PWM), y le damos cierta frecuencia, hacemos la simulación y vemos que espectro de corriente tenemos:

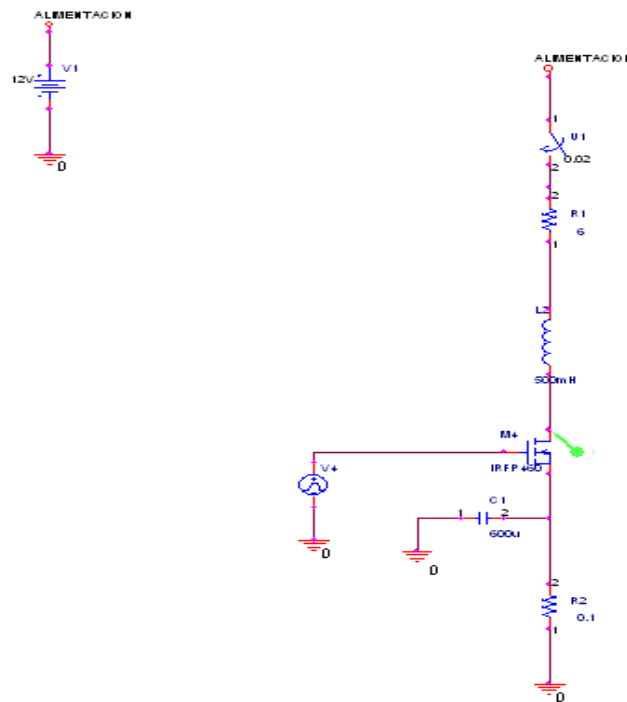


Figura 54: Circuito de potencia 5

Y la grafica que obtenemos es la siguiente:

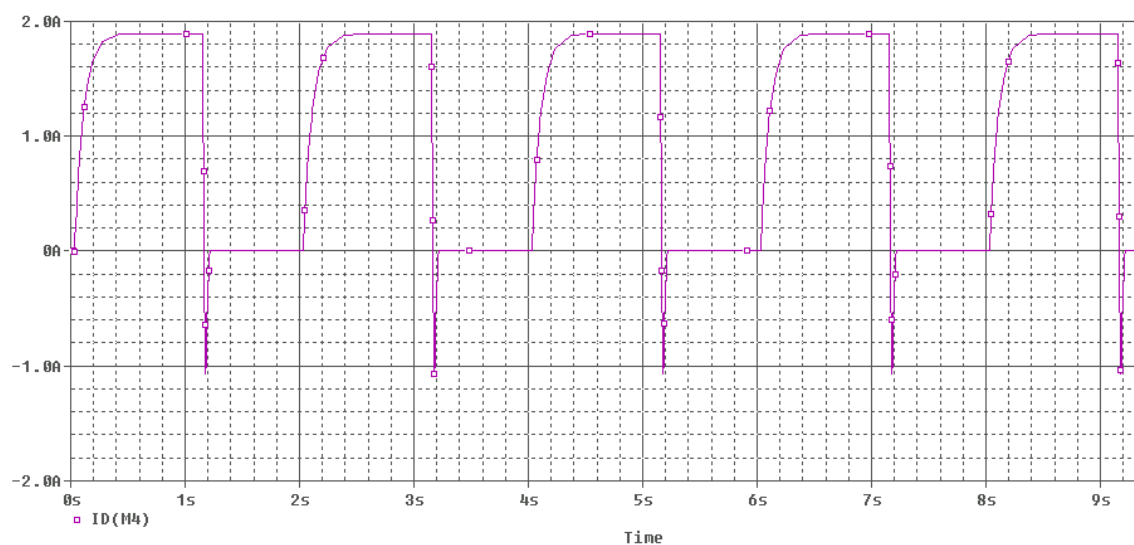


Figura 55: Grafica circuito de potencia 5

Vemos que esos picos de corriente negativa son provocados por la inercia de la bobina, es decir, por su inductancia, obligando al transistor mosfet a circular aunque por su base no exista tensión y por lo tanto tenga la puerta cerrada:

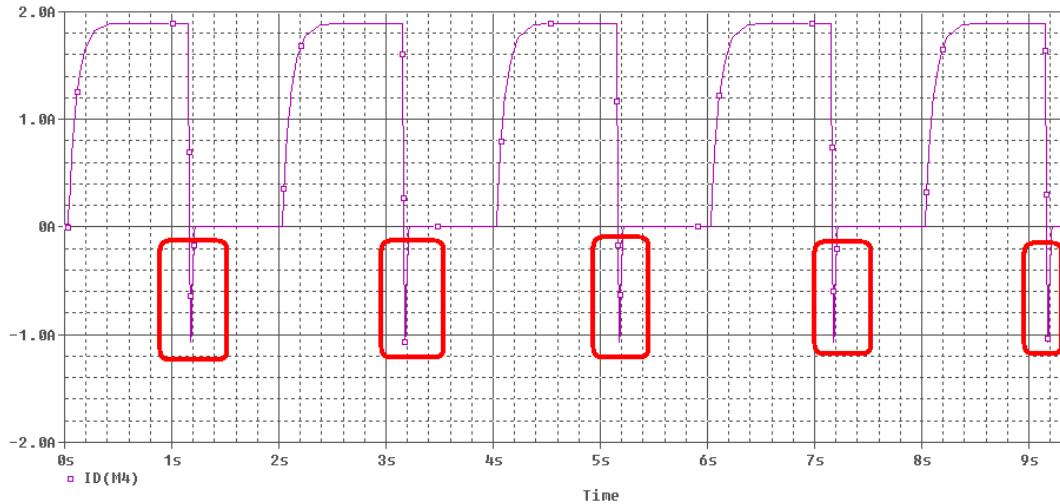


Figura 56: Grafica 2 circuito de potencia 5

Para corregir estos “defectos”, que pueden crear la destrucción del mosfet y un control irregular en el amortiguador es por lo que se usa el diodo en anti-paralelo, para que la corriente busque un camino alternativo más fácil para el paso de corriente y evitar así estos pequeños problemas.

Si ahora le colocamos el diodo en anti-paralelo obtenemos:

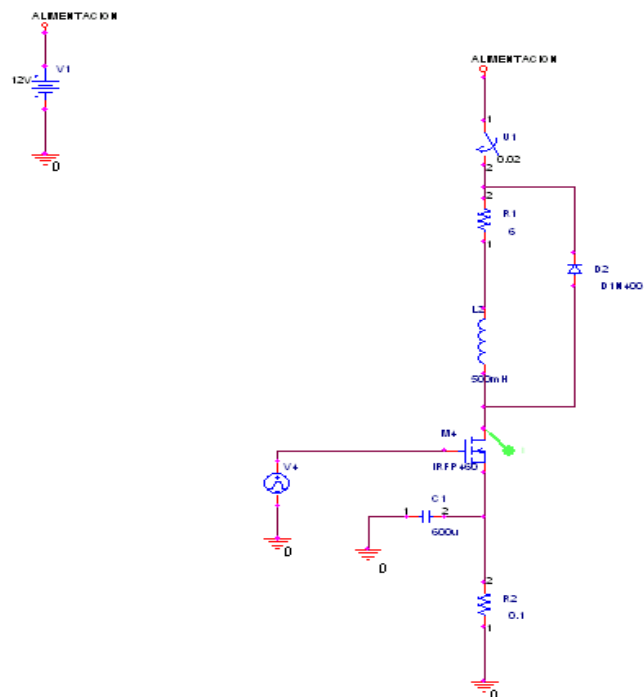


Figura 57: Circuito de potencia 6



Vemos que al poner el diodo en anti-paralelo corregimos ese problema:

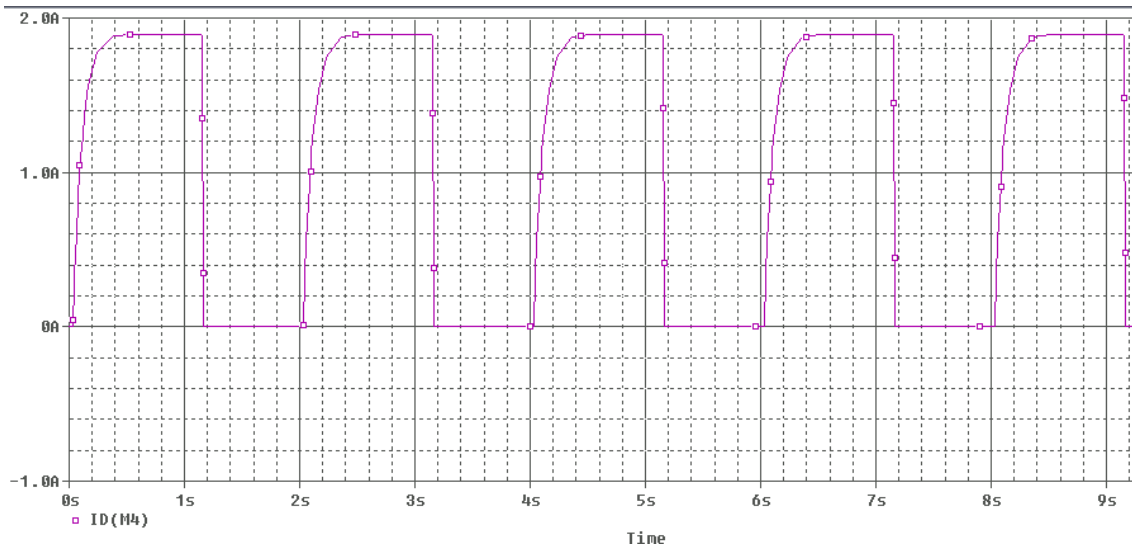


Figura 58: Grafica circuito de potencia 6

Por lo tanto, y debido a la investigación y simulación obtengo que este es el circuito electrónico correspondiente al control que queremos crear sobre el amortiguador magneto-reológico, sin en interruptor al principio, que solo nos sirvió para poder realizar los ensayos.

Ahora solo nos queda realizar el filtrado de la señal:

Comentar que nuestra simulación empieza siempre en 0.02 s, que es cuando cierra el interruptor.

Si creamos un PWM con una oscilación de 1 khz obtenemos:

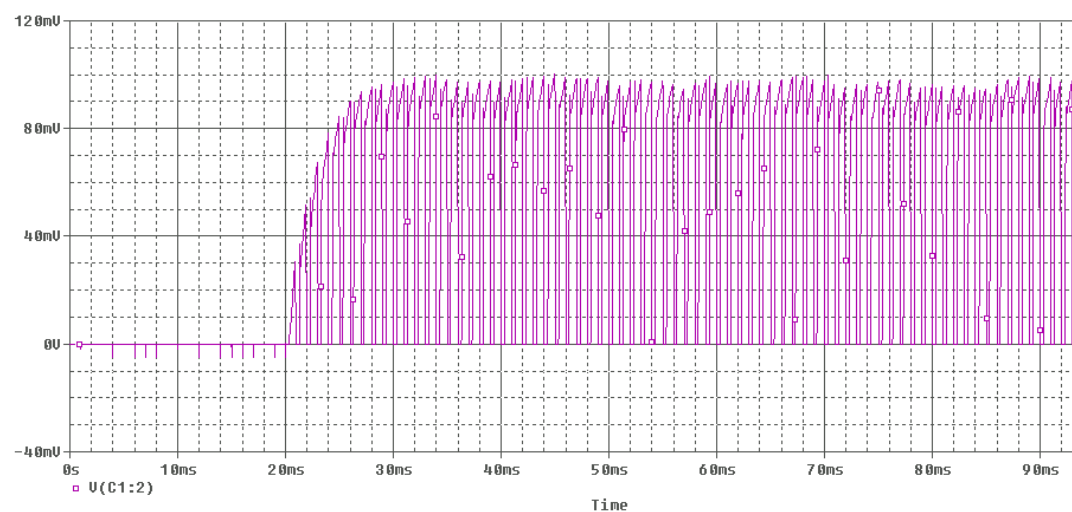


Figura 59: Grafica 2 circuito de potencia 6

Eso lo obtenemos con un condensador de 10uf.

Ahora le ponemos un condensador de 600uf:

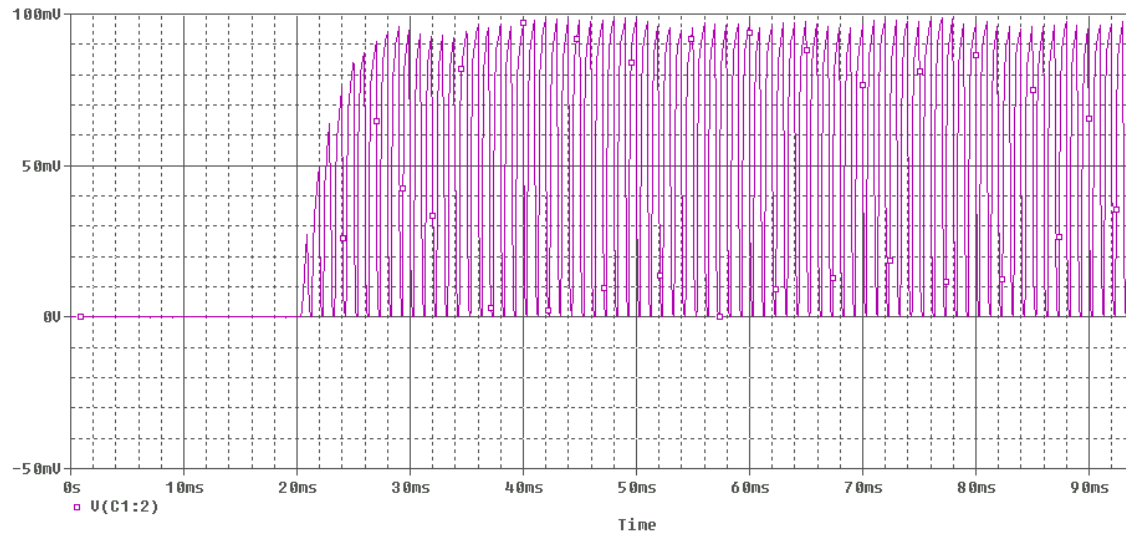


Figura 60: Grafica 3 circuito de potencia 6

Vemos que la señal se modifica, por lo cual, solo nos queda hacer un diseño correcto de filtro para que la señal a amplificar para el microcontrolador ARM7 sea la correcta. Eso pasaremos a describirlo en otro capítulo.

### 3.3. PROGRAMACION DEL MICROCONTROLADOR

#### 3.3.1. IMPLEMENTACION DE LAS FUNCIONES EN EL MICROCONTROLADOR.

El diseño de la lógica de control del amortiguador magneto-reológico se ha realizado a través de un programa diseñado en el software Keil  $\mu$ Vision3. El programa ha sido desarrollado en lenguaje C.

La dificultad del programa no reside en la programación, (ya que tan solo es el control de una interrupción mediante el Timer, el control del ADC por dos canales y un diseño de regulador PID) sino en el estudio, comprensión y configuración del microcontrolador ARM7. Para ello, se expondrá como se deben hacer las configuraciones en este microcontrolador y expondremos todos los cálculos hechos para el diseño del programa.

Las funciones principales de las que consta el programa son las siguientes:

**-Programación de configurado del ARM:** Es el inicio del programa, en el cual decidimos los ciclos de reloj, configuración de los puertos de entrada, salida, timer, etc.

**-Programación del configurado de los vectores de interrupción:** decimos en que parte de la memoria se ubicara la interrupción, duración de la misma y disparos de flanco.

**-Generación del tren de pulsos mediante las interrupciones:** es la parte esencial del programa, en el cual se trabaja constantemente con las señales de entrada para elaborar las respuestas a la salida.

**-PID:** está incluido en el bucle de las interrupciones y va transformando la señal elaborada de salida a una respuesta que nosotros queramos obtener (más rápida o más lenta respecto a un valor de referencia).

**-Bucle de toma de datos del ADC:** es la toma de niveles de tensión en analógica y su transformación en digital. También está incluido en el bucle principal y necesita un tiempo de operación entre toma y toma de niveles que debemos tener en cuenta si queremos que nos dé un resultado correcto.

Una vez descrito por encima las funciones de la lógica implementada en el microcontrolador procederemos a explicar cómo se ha configurado el microcontrolador y como se ha diseñado el programa.

En el ARM7 se puede trabajar con números en base decimal, binaria o hexadecimal, pero para configurar las diferentes funciones o registros se trabaja en base hexadecimal, un ejemplo seria:

TOMCR = 0x0000020B

Para poder interpretar bien esta configuración, primero deberíamos irnos al manual en la parte correspondiente donde se encuentra el registro “T0MCR” y después saber interpretar la configuración.

Podemos decir que detrás del siguiente código “0x” el microcontrolador lo interpreta como que los números que están escritos seguidamente al código están en base hexadecimal y deberemos hacer la siguiente interpretación.

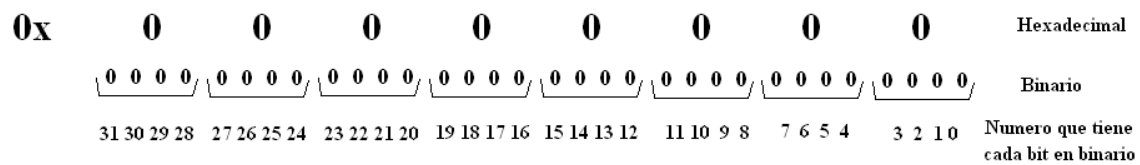


Figura 61: Registro hexadecimal

Vemos su correspondencia de hexadecimal a binario, que cada número en hexadecimal se recoge con cuatro binarios, y que cada binario tiene su número en la palabra completa que es los 32 bits que contiene cada instrucción en los ocho números hexadecimales.

Para poder interpretar y configurar correctamente el microcontrolador deberemos tener muy claro algo tan sencillo como son las tablas de equivalencia en distintas bases

DECIMAL	BINARIO	HEXADECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Figura 62: Tabla conversiones

Hagamos un sencillo ejemplo:

Si cogemos el external match register 0 del timer 0, y decidimos configurarlo como toggle tendremos que proceder de la siguiente manera:

External Match Register (EMR: TIMER0 - T0EMR: 0xE000403C; TIMER1 - T1EMR: 0xE000803C)

EMR	Function	Description	Reset Value
0	External Match 0	This bit reflects the state of output MAT0.0/MAT1.0, whether or not this output is connected to its pin. When a match occurs for MR0, this output of the timer can either toggle, go low, go high, or do nothing. Bits EMR[4:5] control the functionality of this output.	0
1	External Match 1	This bit reflects the state of output MAT0.1/MAT1.1, whether or not this output is connected to its pin. When a match occurs for MR1, this output of the timer can either toggle, go low, go high, or do nothing. Bits EMR[6:7] control the functionality of this output.	0
2	External Match 2	This bit reflects the state of output MAT0.2/MAT1.2, whether or not this output is connected to its pin. When a match occurs for MR2, this output of the timer can either toggle, go low, go high, or do nothing. Bits EMR[8:9] control the functionality of this output.	0
3	External Match 3	This bit reflects the state of output MAT0.3/MAT1.3, whether or not this output is connected to its pin. When a match occurs for MR3, this output of the timer can either toggle, go low, go high, or do nothing. Bits EMR[10:11] control the functionality of this output.	0
5:4	External Match Control 0	Determines the functionality of External Match 0. Table 163 shows the encoding of these bits.	0
7:6	External Match Control 1	Determines the functionality of External Match 1. Table 163 shows the encoding of these bits.	0
9:8	External Match Control 2	Determines the functionality of External Match 2. Table 163 shows the encoding of these bits.	0
11:10	External Match Control 3	Determines the functionality of External Match 3. Table 163 shows the encoding of these bits.	0

External Match Control

EMR[11:10], EMR[9:8], EMR[7:6], or EMR[5:4]	Function
0 0	Do Nothing
0 1	Clear corresponding External Match output to 0 (LOW if pinned out)
1 0	Set corresponding External Match output to 1 (HIGH if pinned out)
1 1	Toggle corresponding External Match output

Figura 63: tabla configuración ARM7

Observamos la tabla de registros y vemos que nos dicen que de los 32 bits, el bit numero 4 y el bit numero 5 son los correspondientes a el external match del timer 0. Y que para configurar este timer 0 como toggle deberé poner los dos bits a 1. Basándonos en la tabla comparativa de arriba, procederé de la siguiente manera:

0x	0	0	0	0	0	0	3	0	Hexadecimal																							
<div><div>0000</div><div>0000</div><div>0000</div><div>0000</div><div>0000</div><div>0000</div><div>0011</div><div>0000</div></div>									Binario																							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Numero que tiene cada bit en binario

Figura 64: Ejemplo hexadecimal

Por lo tanto la instrucción entera que tendré que poner para que esta configuración se haga correctamente será:

T0EMR = 0x00000030

Una vez aclarado el sistema que utilizamos para configurar y programar el microcontrolador, pasaremos a describir la configuración que lleva nuestro programa.

## Funcionamiento.

El funcionamiento de este programa se puede definir por dos grandes partes:

- El vector interrupción que nos generara el PWM.
- La función de toma de datos del ADC por los dos canales seleccionados AIN1 y AIN2.

La clave principal de este programa, es que generamos distintas interrupciones con el MR0, MR1 y MR3 para conseguir el PWM de manera “manual” y que también en uno de los flags de las interrupciones (en MR3) es donde llamamos a la función del ADC para que convierta por los dos canales.

## Creación del PWM.

El reloj interno de la placa MCB2100 funciona a 60 MHz, por lo tanto en base a ella tenemos que hacer funcionar nuestro programa.

La decisión que tome fue que un ciclo de mi programa durara 1 KHz, y que la resolución de toma de muestras o partes en las cuales dividiría el ciclo sería de 1024 posiciones, de ahí partiría el funcionamiento, 1024 sería el total y el porcentaje de posiciones con el Duty cycle en alto y el porcentaje en bajo variara atendiendo a las circunstancias.

$$PWM \rightarrow 1 \text{ KHz} \rightarrow 0,001 \text{ sg}$$

$$Duty \text{ cycle} \rightarrow 0\% \text{ a } 100\% \rightarrow \text{en } 1024 \text{ posiciones}$$

De aquí obtenemos que nuestro tick en el clk del reloj del MCB2100 sería de:

$$Tick = \frac{0,001 \text{ sg}}{1024 \text{ posiciones}} = 9,765 * 10^{-7}$$

Este cálculo nos dice que una posición equivale a  $9,765 * 10^{-7}$  segundos en el tiempo

Ahora vemos los segundos del reloj interno

$$\frac{1}{60 * 10^6} = 1,666 * 10^{-8} \text{ sg}$$

Con nuestros valores calculados procedemos a hacer nuestro pre-escalado

$$Pre - \text{escalado} = \frac{9,765 * 10^{-7}}{\frac{1}{60 * 10^6}} = 58,58$$

Lo que podemos aproximar por 59.  
Y lo metemos en el programa:

```
VPBDIV = 0x00000001; // igualo cclk=pcclk
TOPR = 0x0000003B;    // pre escalado a 59 tick (hecho con cálculos) con 60 Mhz
```

Ahora que ya tenemos nuestro ciclo de programa sincronizado con el reloj del MCB2100, procederemos a diseñar nuestras interrupciones.

El diseño es el siguiente, tenemos un timer 0 que nos marca las 1024 posiciones del ciclo, un MAT 1 que nos dará el flanco de bajada del ciclo en el instante del mismo que nosotros configuremos y un MAT 0 que nos dará el flanco de subida en cada nuevo ciclo.

Esta sería la explicación teórica de nuestro sistema de control del PWM

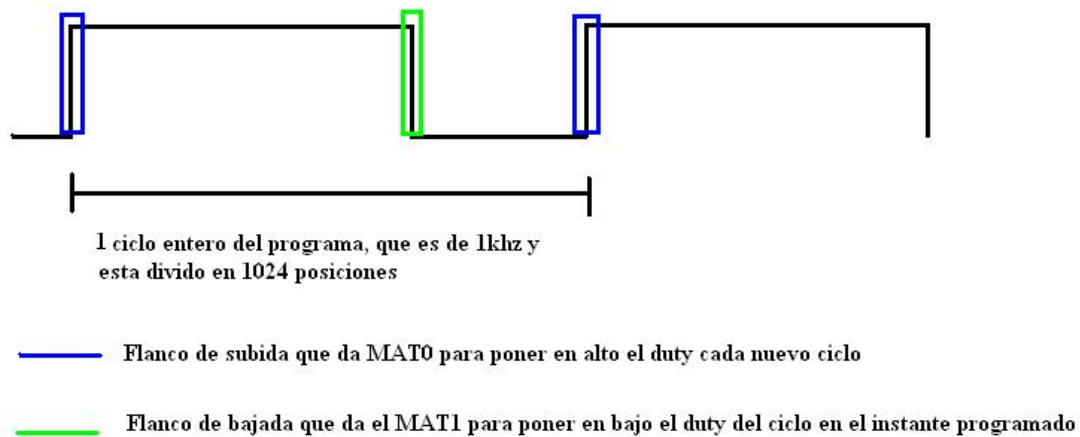


Figura 65: PWM generado

Y de una manera simulada en nuestro programa quedaría como sigue:

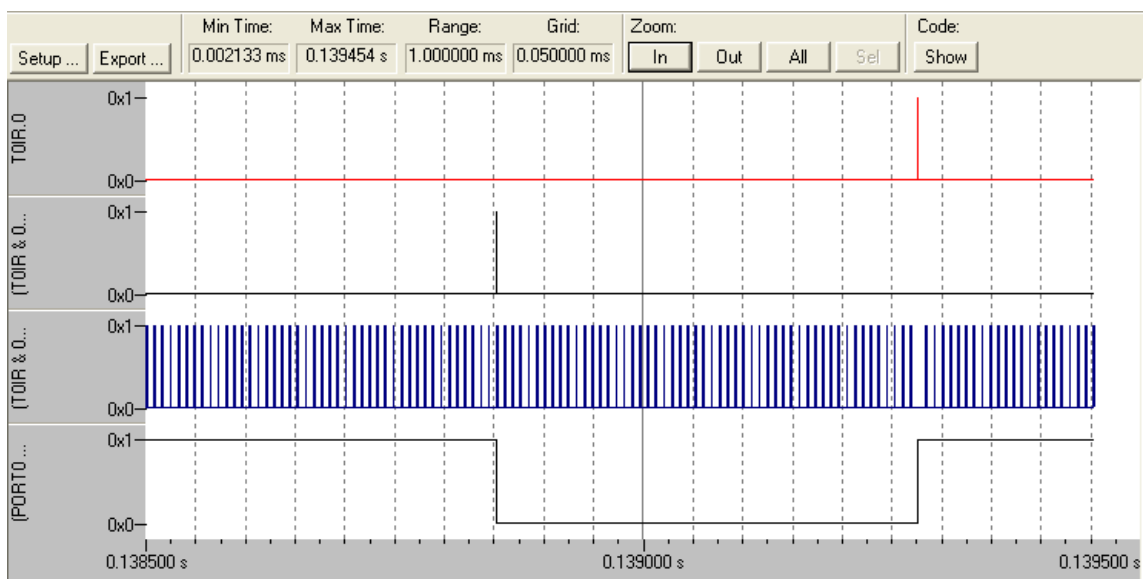


Figura 66: Simulación señales programa

Si vamos siguiendo desde arriba las señales, nos encontramos con:

- la primera señal en rojo es MAT0 que da el flanco cada nuevo ciclo para poner en alto el duty cycle del PWM y a 1 la salida hacia el mosfet.
- La segunda señal en negra es el MAT1 que da el flanco para poner en bajo el duty cycle del PWM y pondrá a 0 la salida hacia el mosfet.
- La tercera señal en azul es el MAT3, que va contando la toma de muestras del ADC, que explicaremos más adelante
- La cuarta señal en negro también, es el duty cycle que generamos, que se pone en alto o en bajo según le lleguen los flags de las señales MAT0 y MAT1.

## Creación de la función del ADC

Ahora procederemos a explicar cómo funciona el ADC y la función creada para su toma de datos.

De una manera paralela e independiente a nuestra interrupción del PWM se crea una toma de datos en la función cada vez que salta el flag del MR3. Debemos aclarar un concepto importante, y es que la función del ADC se ejecutara independientemente del programa cada vez que el ADC termine una conversión, pero la toma de muestras de nuestro ADC será controladas por la interrupción del Timer 0 mediante el MAT3, por lo tanto debe existir tiempo suficiente entre muestra y muestra que tome el MAT3 y el tiempo que dura cada conversión, de lo contrario las conversiones se solaparan y no funcionarían bien. Estos factores son los que debemos tener en cuenta para hacer un buen muestreo.

Sabemos que tenemos un ciclo que trabaja a 1 KHz , es decir, que dura 0,001 segundos y necesitamos hacer un buen muestreo de este mismo para conseguir una valor representativo real del valor del duty cycle.

La idea del ADC es tomar un número de muestras de todo el ciclo, y contar todas las muestras, y hacer una media de todas esas muestras para saber el número de tick que dura el duty cycle del máximo posible de un ciclo.

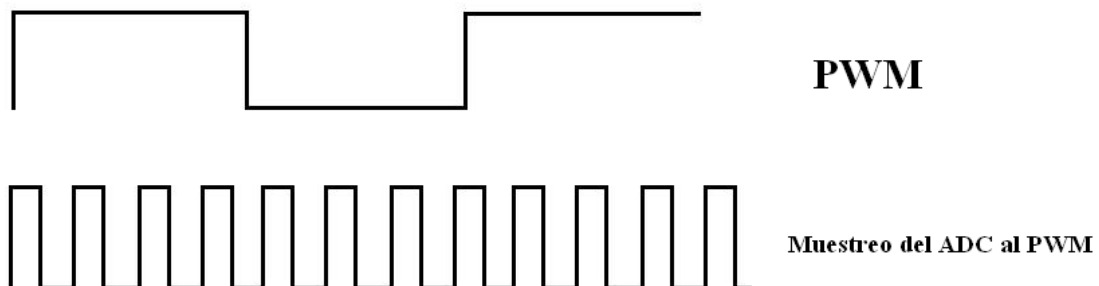


Figura 67: Captura del ADC

Para controlar nuestro muestreo lo haremos con el MAT3, atendiendo a cuando se cumpla el flag del MAT3 el ADC saltará y tomara valor de la señal



Tenemos que observar que nos encontramos con dos aspectos a tener en cuenta:

- El primero es saber a qué frecuencia muestreara el ADC el ciclo de 1KHz del PWM
- El segundo es ajustar la rapidez de la conversión del ADC para que pueda trabajar bien y no se solapen unas conversiones con otras.

La frecuencia de muestreo elegida, en base a los resultados que se obtuvo simulando fue de 15 muestras por ciclo que tomara nuestro ADC, nos dará una precisión aceptable y al ADC le dará tiempo a hacer la conversión de analógico a digital bien.

La frecuencia de trabajo de nuestro ADC será de 3 MHz, donde trabajando en este rango puede hacer las conversiones a tiempo.

Nuestro ADC realizara trabajara con dos canales, el AIN0 y el AIN1; por lo tanto tendrá que realizar una conversión para cada canal en cada muestreo:

- AIN0: esta conversión la realizara 15 veces por ciclo y actualizara continuamente los valores que está teniendo el potenciómetro en ese momento.
- AIN1: esta conversión la realizara 15 veces por ciclo y será tomada de los valores del PWM del amortiguador para posteriormente con todos ellos hacer una media para calcular el nuevo duty cycle.

Por lo tanto deberemos tener muy en cuenta estas dos conversiones para que al ADC le dé tiempo a realizar todo el trabajo pedido.

Los cálculos que realizamos para saber si teóricamente al ADC le daría tiempo a realizar todas las conversiones correctamente fueron las siguientes:

Calculamos un valor proporcional y aceptable de dividir en 16 periodos el ciclo de 1khz, por lo tanto sabemos que entre toma de muestras y la siguiente debe haber un tiempo de:

$$Tp: \frac{0,001}{16} = 0,0000625 \text{ sg}$$

Este es el tiempo que dispone el ADC para hacer las dos conversiones a 10 bits, si el ADC trabaja a 3 Mhz, entonces:

$$Tclk: \frac{1}{3} \text{ MHz} = 3,33 * 10^{-7} \text{ sg}$$

Por estar el ADC a 10 bits, necesita 11 ciclos para realizar la conversión, por lo que el tiempo total será:

$$Tconv: (3,33 * 10^{-7} \text{ sg}) * 11 = 3,33 * 10^{-6} \text{ sg}$$

Sabemos también que debe realizar dos conversiones, una para el AIN0 y otra para el AIN1, por lo tanto el tiempo total de conversión será:

$$Ttconv : (3,33 * 10^{-6}) * 2 = 0,00000666 \text{ sg}$$

Y ahora analizamos y vemos que :

$T_p \gg T_t \text{ conv; ya que } 0,0000625\text{sg} \gg 0,00000666 \text{ sg}$

Cabe resaltar que de las 16 interrupciones nosotros en realidad tomamos 15 muestras, una menos, ya que el MR3 produce la primera interrupción pasada la interrupción general que produce el Timer 0.

Por último podemos diferenciar los tres bloques de actuación del ADC en el programa:

### **El momento en el cual se activa el ADC mediante el flag del MR3:**

```
if(T0IR&0x08) //MR3
{
    medidas();           //Tomo medidas del ADC
    T0MR3 = T0MR3 + 64;  // actualizo el ciclo del ADC en otros 64 tick mas
    T0IR = 0x08;         //borro flag INT
}
```

### **El bucle de conversión de mi ADC:**

```
static unsigned short ADC_Read(unsigned char ch)
{
    unsigned int i;
    ADCR = 0x00201300 | ch;  //Inicializo y selecciono canal
    ADCR |= 0x01000000;      // Start A/D Conversion
    do
    {
        i = ADDR;            // Leer A/D
    } while ((i & 0x80000000) == 0); // Espero conversion
    return (i >> 6) & 0x03FF;  // bit 6:15 es el bit 10 AD
}
```

### **La función que pide que haga medidas por los dos canales:**

```
int medidas(void)
{
    val_POT = (ADC_Read(1));          // convierto canal AIN0
    val_ADC = val_ADC + (ADC_Read(2)); // convierto canal AIN1
}
```

Por último, y una vez explicado el funcionamiento del programa, nos queda saber cuáles serán nuestras salidas y entradas del ARM7 hacia el exterior.

-PWM hacia el transistor: Salida P0.12

-Toma de PWM hacia el ADC: Entrada P0.28

-Toma de dato valor de referencia del potenciómetro hacia el PID: Entrada P0.27

### 3.3.2. DIAGRAMAS DE FLUJO

Para probar si nuestro programa funciona correctamente y el amortiguador obedece se hizo una primera prueba en lazo abierto



Figura 68: Lazo abierto

Después, viendo que nuestro sistema operaba bien, se procedió a probar el sistema completo, quedando en lazo cerrado como muestra la figura de abajo.

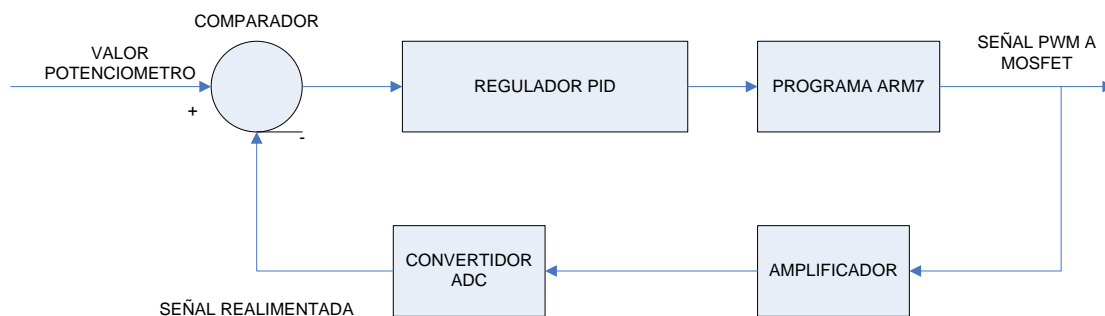
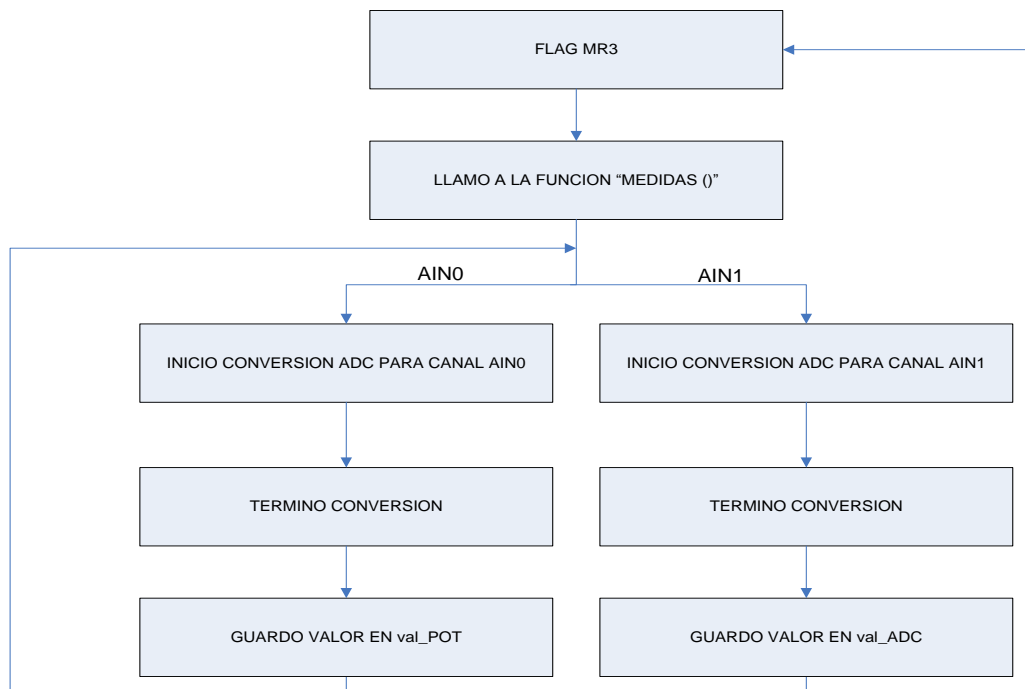
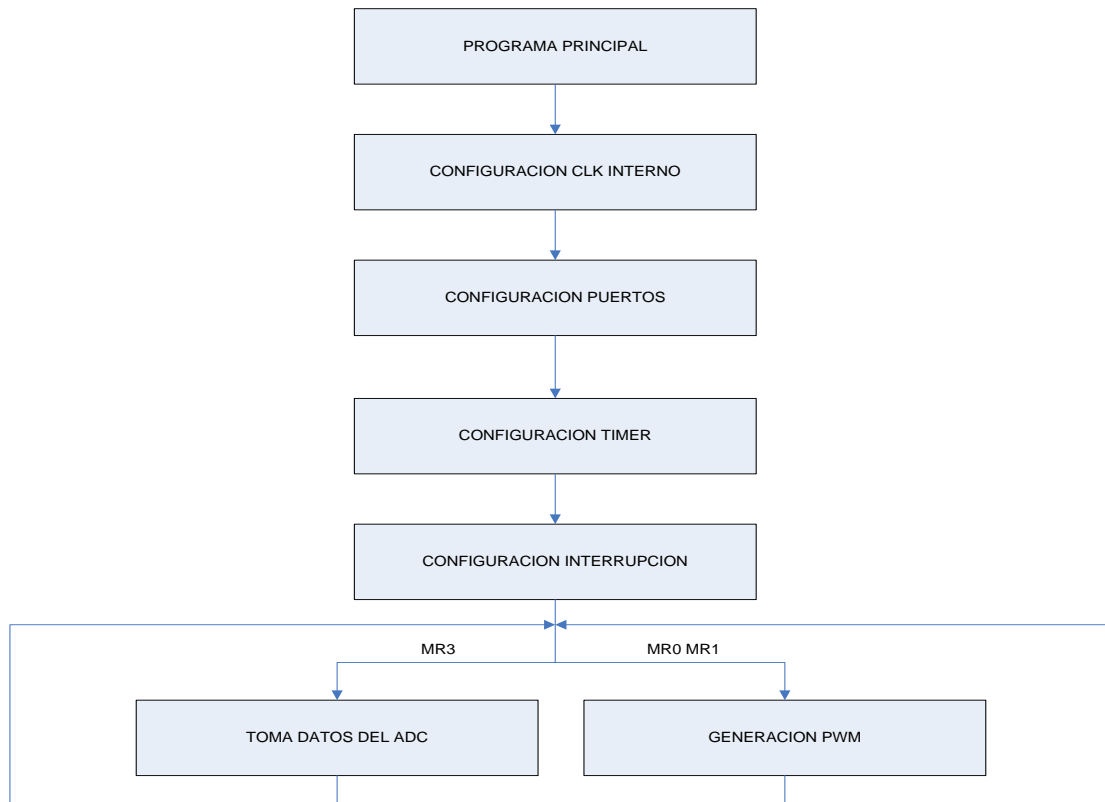


Figura 69: Lazo cerrado

Mediante diagramas de flujos podemos representar también el funcionamiento del programa diseñado para el control del PWM para el amortiguador magneto-reológico.



## INTERRUPCION DEL TIMER 0

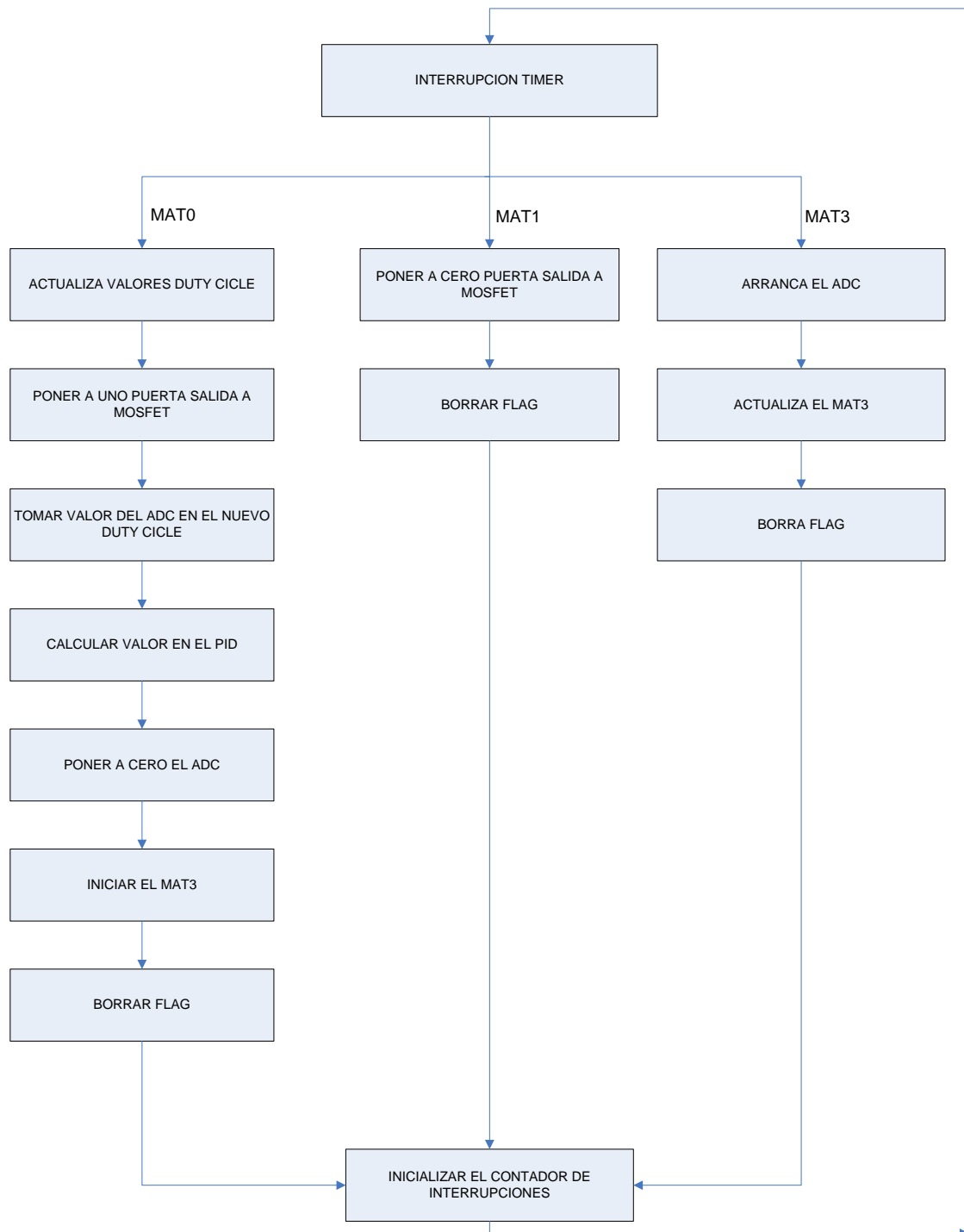


Figura 72: Diagrama flujo interrupción Timer 0

## Página 69

### 3.4.2. AJUSTE CIRCUITO DE POTENCIA

Concebido nuestro circuito de potencia para controlar el amortiguador magneto-reológico, solo nos queda depurar el diseño para hacer un control óptimo de la señal.

Lo primero que debemos hacer, es crear un sistema para que la señal nos salga limpia de ruidos y sea una señal buena para ser analizada (siguiendo las recomendaciones del fabricante para el uso del ADC).

Una vez observado el carácter inductivo del sistema a controlar (amortiguador magneto-reológico) sabemos que ante los cambios de señal que producimos en el circuito (PWM) la respuesta será lenta debido a la fuerza contra-electromotriz auto-inducida, por lo que para frecuencias elevadas, apenas generaremos PWM, sino una señal prácticamente continua, ya que el retardo que producen dichas fuerzas y la velocidad con que el cambio de señal en el PWM se produce hacen que apenas a la señal la de tiempo a variar, por lo tanto nos será inservible.

Calcularemos un filtro para frecuencias máximas de 10 KHz, es decir, crearemos un filtro paso-bajo con frecuencia de corte de 10 KHz, y analizaremos si este filtro nos es suficiente para eliminar el ruido o tenemos que restringirlo a frecuencias inferiores.

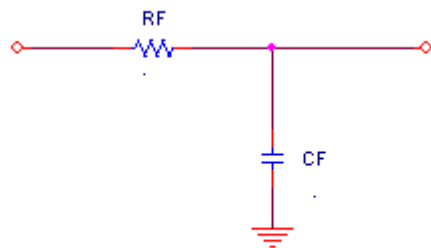


Figura 74: Filtro RC

Nuestro filtro tendrá la característica de que las frecuencias superiores a 10.000 Hz no las dejen pasar, las eliminara.

De la ecuación:  $\omega_c = \frac{1}{R \cdot C}$

Con  $\omega_c = 2\pi f$ ;

$$\omega_c = 2 \cdot \pi \cdot 10000 = 62800$$

Y ahora imponemos que el condensador del filtro sea de 10 $\mu$ F.

$$\omega_c = \frac{1}{R \cdot 10\mu F}$$

$$\omega_c = 62800$$

$$62800 * 10\mu F = \frac{1}{R}$$

$$R = 1,6 \text{ ohm}$$

Por lo tanto:  $C_F = 10\mu F$  y  $R_F = 1.6\Omega$

Una vez calculado nuestro filtro paso bajo, lo que tenemos que hacer es un diseño para la amplificación de la señal en tensión que nos llegara, ya que hemos visto más para atrás, que la tensión que nos llegara a nuestro sistema será de:

$$0.25 * I_{max} = V_{max}, \text{ por lo que}$$

$$0.25 * 2 = 0.5 \text{ V}; \text{ vemos que es un voltaje pequeño,}$$

Procederemos entonces al diseño de la amplificación de esta señal, una vez que nos llegue limpia de ruidos gracias al filtro paso bajo.

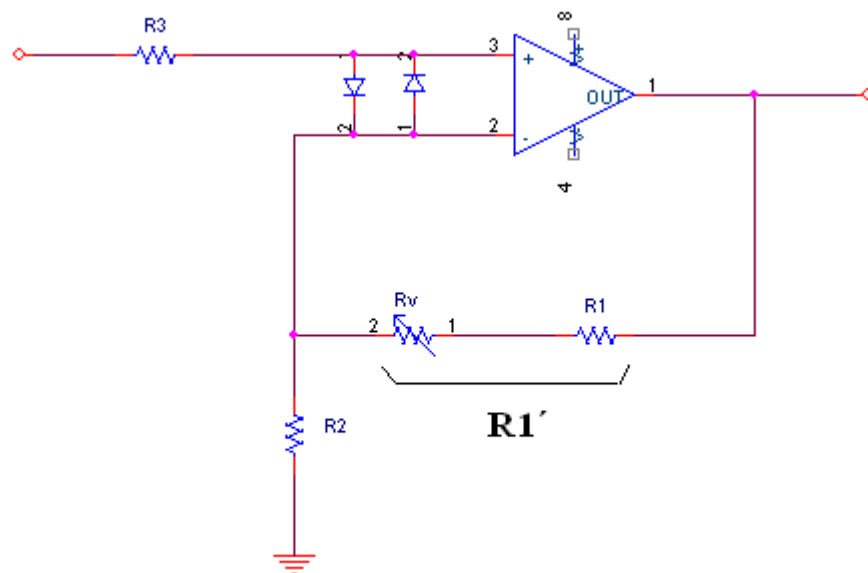


Figura 75: Diseño Operacional

El resistor variable tendrá un valor máximo de 200 K pero para los cálculos, los realizaremos con la mitad, es decir, con 100 K.

El valor de R2 será de 100 K, será un valor que imponemos como criterio.

Teniendo en cuenta condiciones reales, el máximo de intensidad que se tendrá es de 3 A Por lo que si realizamos los cálculos:

$$\frac{0,75}{0,25} = 3 \text{ Amperios}$$



$$V_{\max} = 0,75 \text{ V}$$

Nuestro valor amplificado queremos que alcance valores máximos de:

$$V_{\text{amplificada}} = 3 \text{ V}$$

Por lo que la ecuación de nuestro amplificador no inversor será de:

$$\left(1 + \frac{R1'}{R2}\right) = \frac{3V}{0,75V}$$

El valor de  $R1'$  será de:

$$R1' = R_v + R1$$

Si seguimos calculando y nos vamos a la ecuación anterior encontramos que:

$$\left(1 + \frac{R1'}{R2}\right) = 4 ; \text{ de aquí despejamos sabiendo que } R2 = 100K$$

Y obtenemos que:

$R1' = 300 \text{ K}$ , sabiendo el valor de  $R1'$  despejamos el valor de  $R1$ :

$$R1 = 300 \text{ K} - 100 \text{ K} = 200 \text{ K}$$

Ya tenemos el valor de  $R1$ ; ahora solo nos queda calcular el valor de  $R3$ :

$$R3 = R2 // R1'$$

$$R3 = \frac{300K * 100K}{300K + 100K} = \frac{30000M}{400K}$$

$$R3 = 75 \text{ K}$$

Por lo tanto obtenemos que:

$$R1 = 200 \text{ K} ;$$

$$R_v = 200 \text{ K de max.};$$

$$R2 = 100 \text{ K};$$

$$R3 = 75 \text{ K};$$

Una vez obtenido todos los valores, solo nos queda simular y ver qué resultados obtenemos. Es ahora que mediante la herramienta Pspice de Orcad nos disponemos a simular barridos y observar resultados y actuación del filtro calculado.

## BARRIDO 1: A 1KHZ Y DISTINTOS ANCHOS DE PULSO

BARRIDO SIN FILTRO:

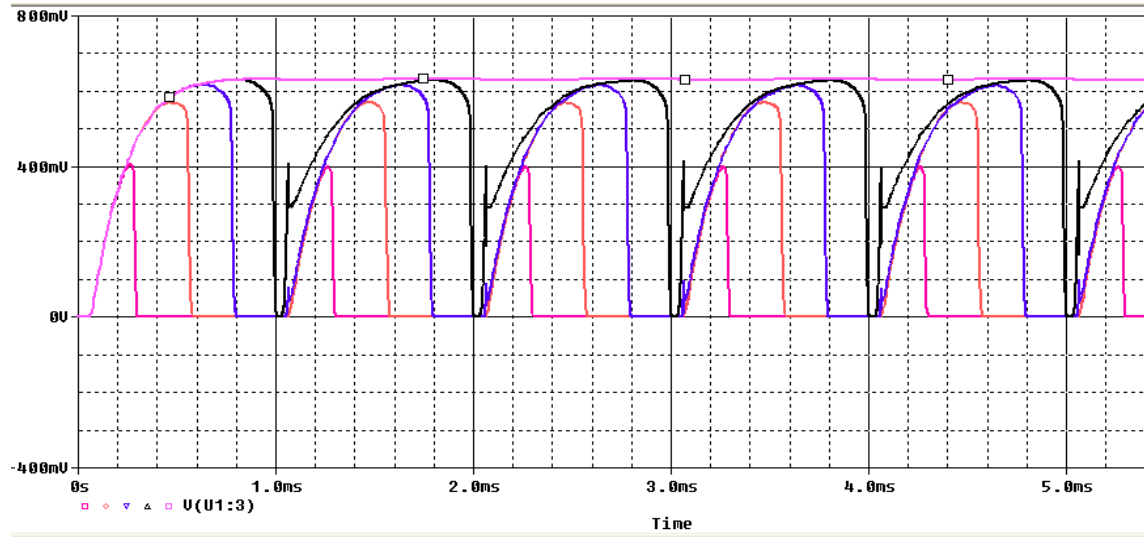


Figura 76: Barrido 1 KHz: Anchos de pulso (10%, 30%, 60%, 80%, 100%) Sin filtro

BARRIDO CON FILTRO:

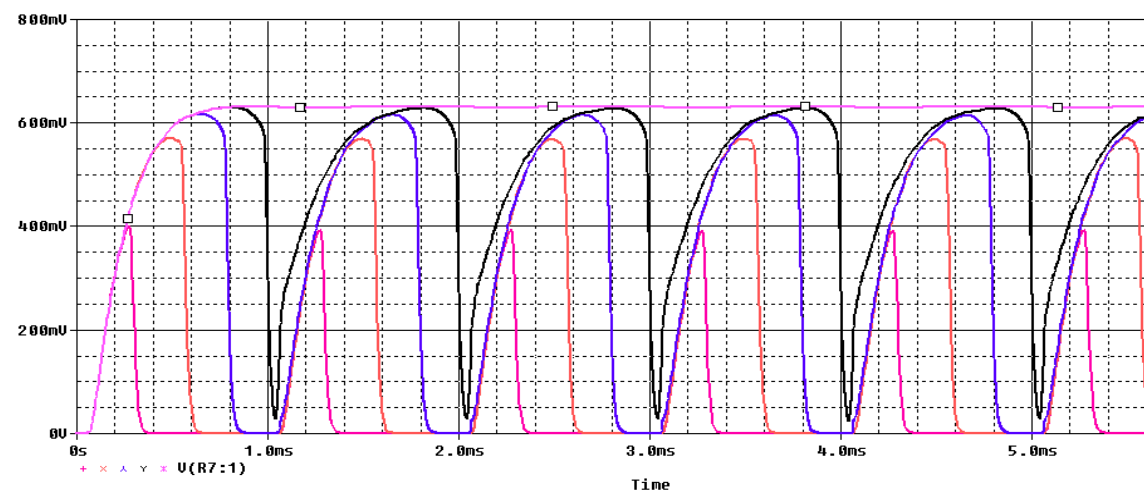


Figura 77: Barrido 1 KHz: Anchos de pulso (10%, 30%, 60%, 80%, 100%) Con filtro

Vemos que el filtro elimina las distorsiones en la señal, por lo tanto está bien diseñado y hace el sistema más fácilmente tratable.

## BARRIDO SEÑAL AMPLIFICADA:

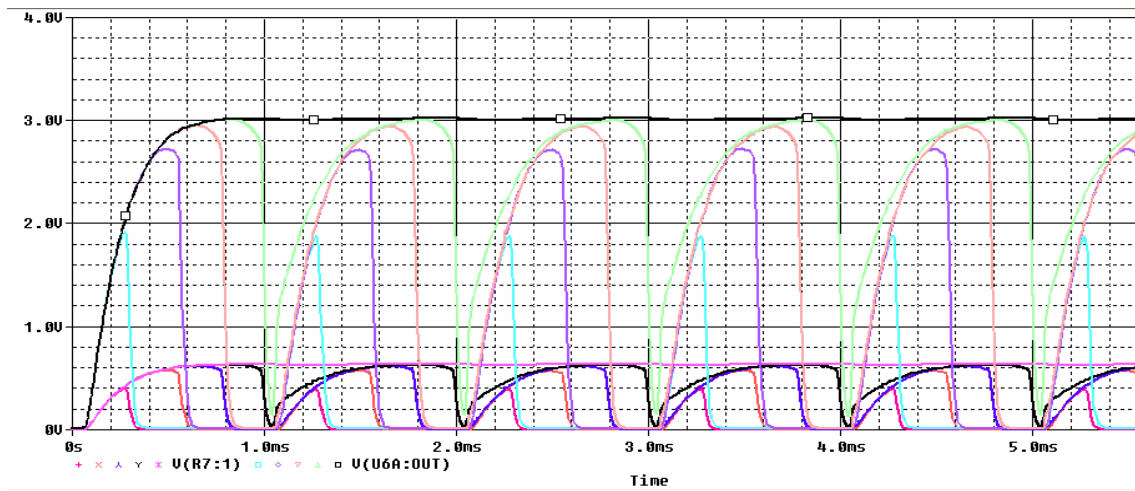


Figura 78: Barrido 1 KHz: Anchos de pulso (10%, 30%, 50%, 70%, 90%) Con filtro. Señal Amplificada

Aquí vemos que obtenemos una buena señal filtrada y amplificada.

Después en las siguientes simulaciones, con esos mismos valores R y L hemos hecho un barrido con el mismo ancho de pulso, pero para diferentes frecuencias. En este punto se puede decir que estas graficas no son reales, son aproximadas, ya que R y L varían con la frecuencia y en estas simulaciones la frecuencia variaba, pero R y L no variaba, lo que nos lleva a deducir que los resultados son aproximados y no exactos, pero nos es un método de estudio valido y aproximado. Los resultados fueron los siguientes:

## BARRIDO 2: A 1/2 DE CICLO Y A DIFERENTES FRECUENCIAS DE 1K HASTA 20KHZ

### BARRIDO SIN FILTRO:

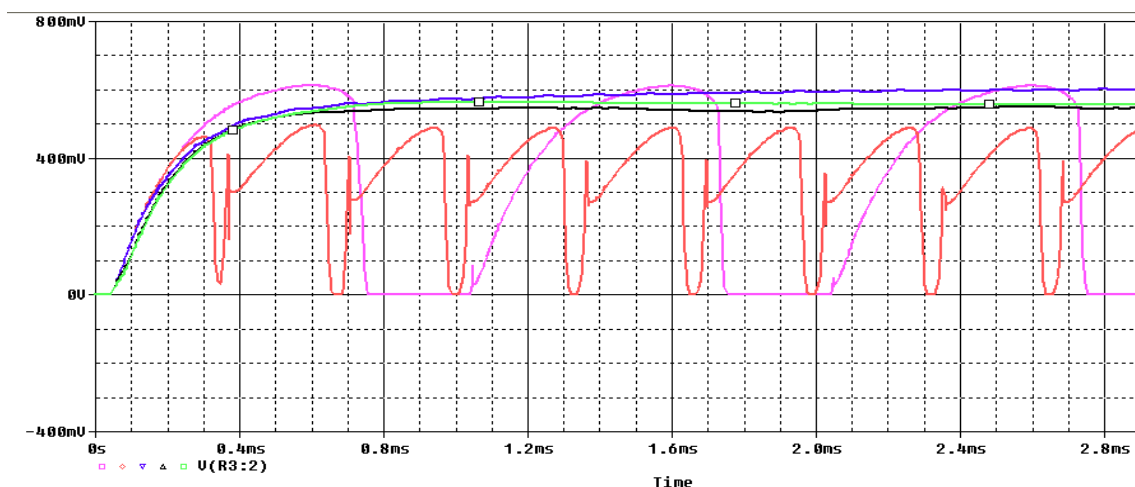


Figura 79: Barrido Ancho de pulso al 50% con frecuencias de 1 KHz, 3 KHz, 7KHz, 15 KHz, 20 KHz. Sin filtro

## BARRIDO CON FILTRO:

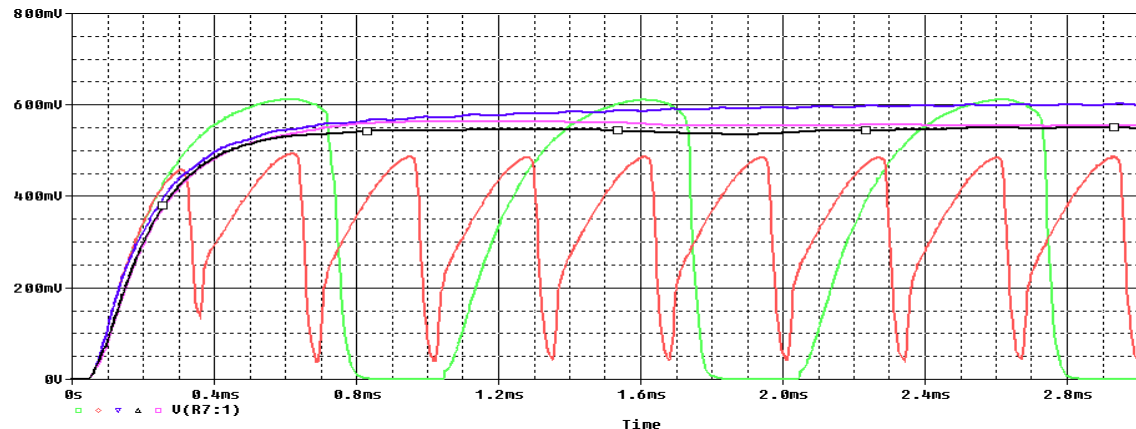


Figura 80: Barrido Ancho de pulso al 50% con frecuencias de 1 KHz, 3 KHz, 7KHz, 15 KHz, 20 KHz. Con filtro

## BARRIDO SEÑAL AMPLIFICADA:

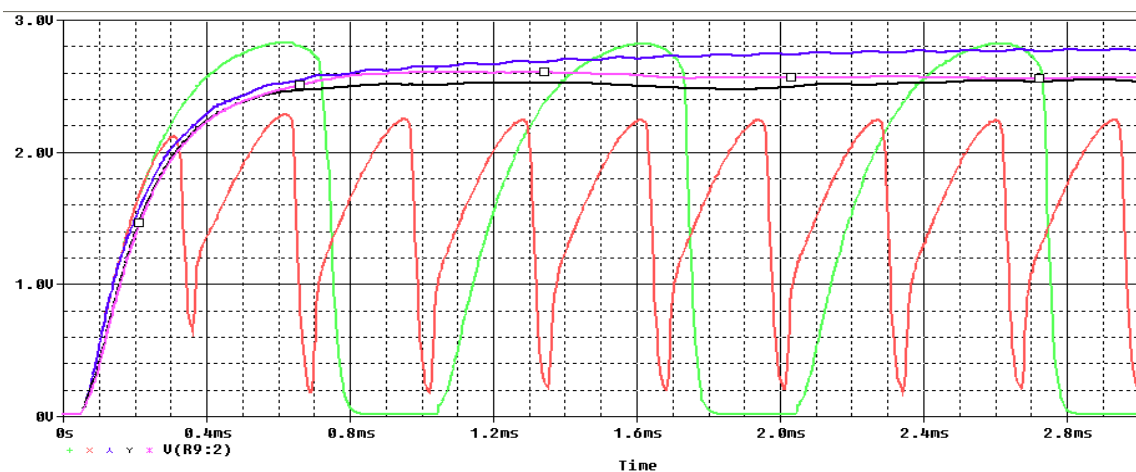


Figura 81: Barrido Ancho de pulso al 50% con frecuencias de 1 KHz, 3 KHz, 7KHz, 15 KHz, 20 KHz. Con filtro y Amplificada

Vemos que el filtro nos permite amplificar una señal con muy buenos resultados. Pero haciendo un estudio sobre como variaba y en qué proporción los parámetros R y L del amortiguador con la frecuencia. Podemos ver en la figura 82 que forma tienen las graficas que describen el comportamiento de un conductor y su impedancia frente a la variación de la frecuencia.

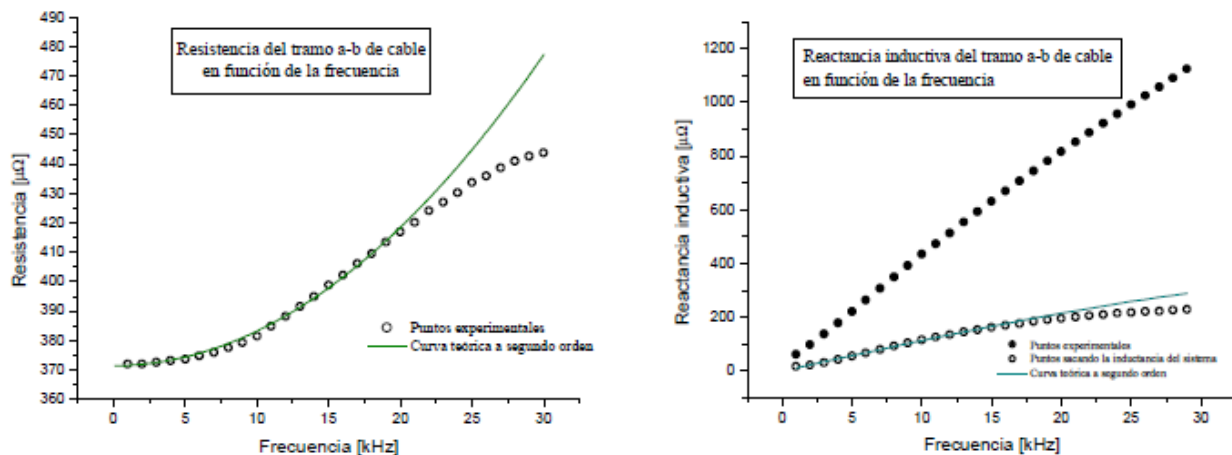


Figura 82: Impedancia variable

Vemos como es la variación de la impedancia de un cable con carácter inductivo mediante la variación de la frecuencia. Esto nos es útil para hacer el estudio de la variación de la impedancia en el amortiguador magneto-reológico.

Haciendo un estudio y caracterización del amortiguador obtenemos los siguientes resultados en el laboratorio:

Valores de impedancia del amortiguador magneto-reológico		
Frecuencia	Valores de R	Valores de L
100 Hz	2 ohmios	2,25 mili Henrios
1 KHz	5,03 ohmios	774 micro Henrios
10 KHz	15,16 ohmios	283,8 micro Henrios

Con toda esta información y basados en la observación, llegamos a la conclusión de que la curva que se describe en la base teórica se podría aproximar a una recta. Estableciendo una proporción de aproximación lineal y con los datos que obtuvimos de las pruebas en el laboratorio sacamos la ecuación de la recta correspondiente a esos valores.

Con los puntos de L y los puntos de R, sus ecuaciones de recta correspondientes:

Para la resistencia obtenemos en función de la frecuencia la siguiente.

$$y = 0.001125x + 3.9$$

Para la inductancia de L obtenemos en función de la frecuencia esta recta.

$$y = (-0.05446 \cdot 10^{-6})x + 828.4 \cdot 10^{-6}$$

Estas rectas las metemos como datos en los valores R y L y obtenemos una simulación muy buena y muy real al realizar el barrido.

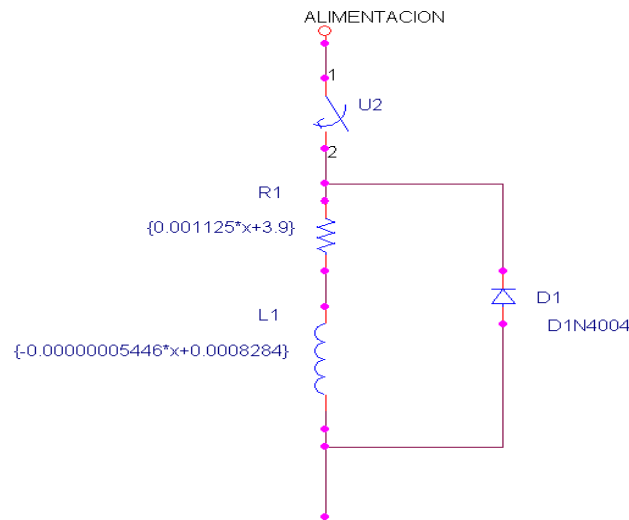


Figura 83: Impedancia variable esquemáticos

Los barridos realizados con la impedancia variable en función de la frecuencia fueron los siguientes:

**BARRIDO 3: BARRIDO CON UN MISMO ANCHO DE PULSO, PERO DISTINTAS FRECUENCIAS, Y CON IMPEDANCIA VARIABLE.**

**BARRIDO SIN FILTRO:**

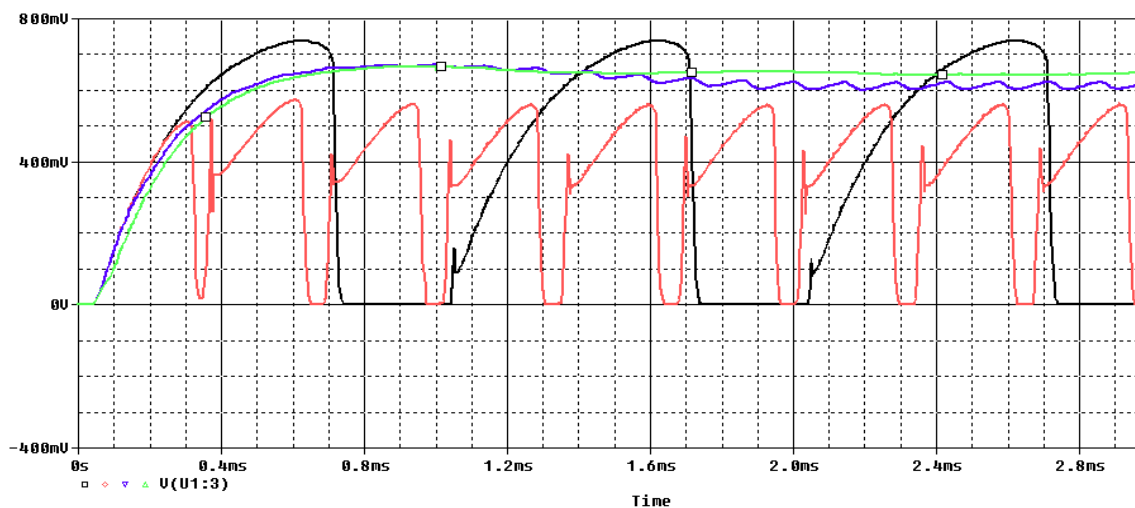


Figura 84: Barrido Ancho de pulso al 50% con frecuencias de 1 KHz, 3 KHz, 7KHz, 15 KHz. Sin filtro

## BARRIDO CON FILTRO:

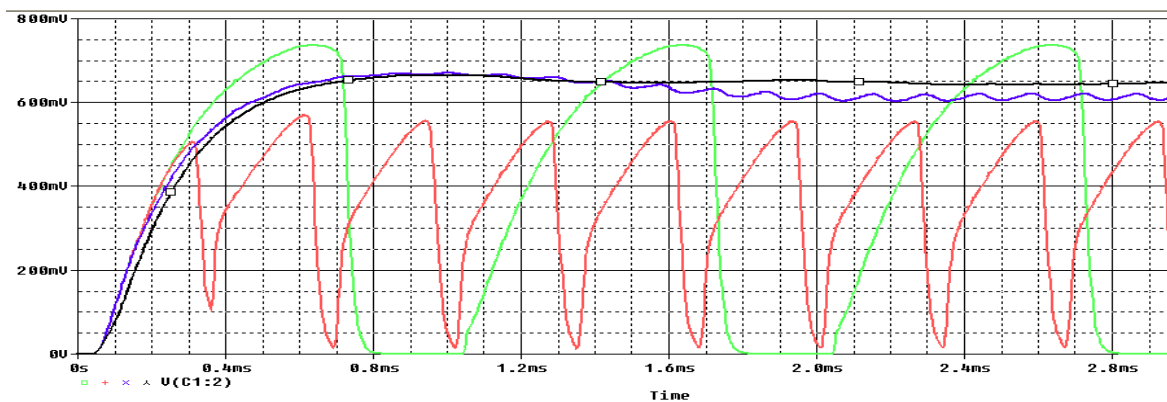


Figura 85: Barrido Ancho de pulso al 50% con frecuencias de 1 KHz, 3 KHz, 7KHz, 15 KHz. Con filtro

## BARRIDO CON AMPLIFICADOR:

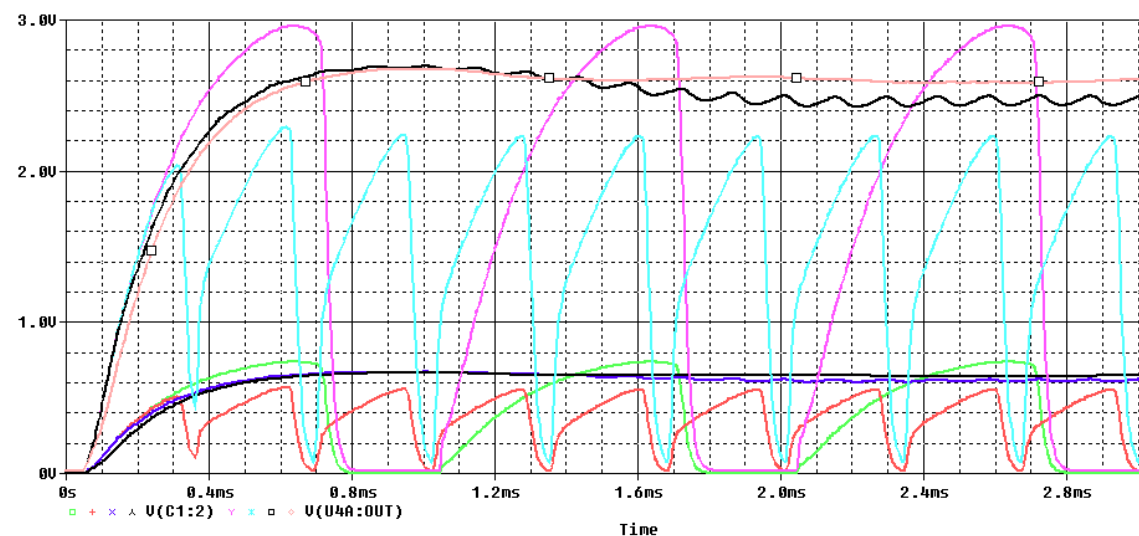


Figura 86: Barrido Ancho de pulso al 50% con frecuencias de 1 KHz, 3 KHz, 7KHz, 15 KHz. Con filtro y Amplificada

De los resultados obtenidos en las simulaciones podemos llegar a la conclusión de que los cálculos y aproximaciones realizadas para el diseño del circuito de potencia son validos.

### 3.4.3. TRATAMIENTO DE LA SEÑAL

Una vez simulados distintos anchos de pulsos con distintas frecuencias para ver si realmente actuaba el circuito de potencia según lo diseñado y calculado, ahora solo nos queda hacer un estudio sobre cuál es la frecuencia optima de trabajo para generar nuestro ancho de pulso y que el sistema funcione de una manera controlable.

Este es nuestro sistema de potencia ya ajustado.

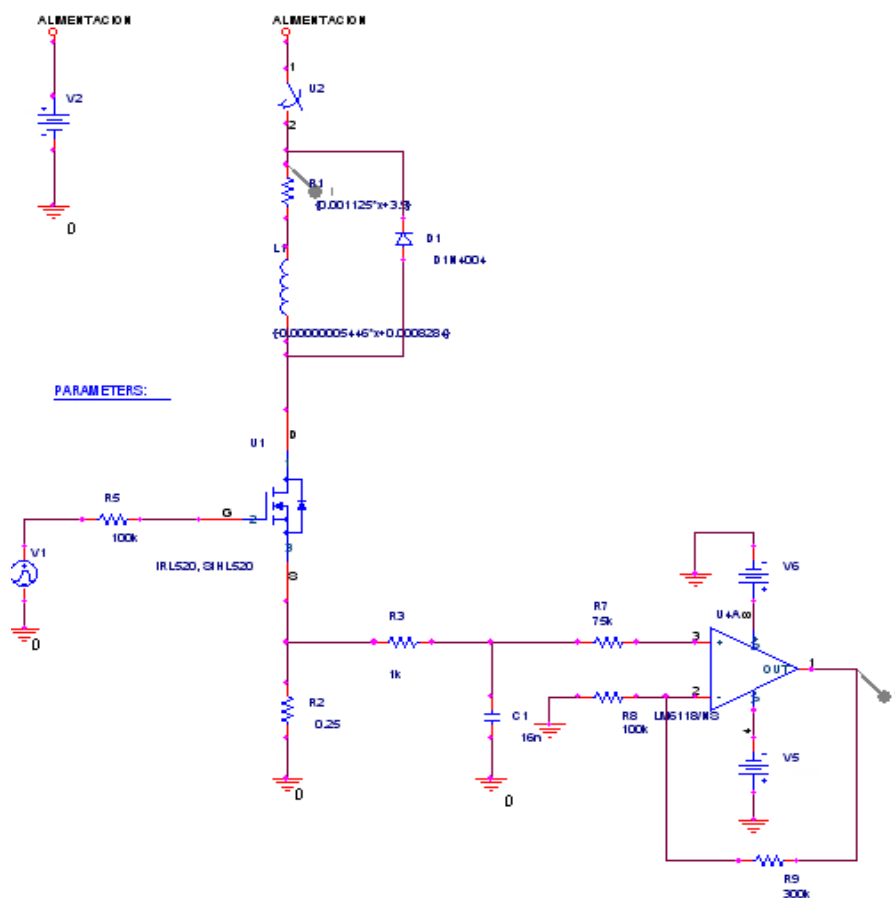


Figura 87: Simulación Circuito control potencia

Ahora procederemos a hacer distintos barridos variando el porcentaje de duty cycle del PWM y ver el comportamiento y la forma de onda de la señal que se recoge para las diferentes frecuencias analizadas

Como es un control mediante PWM nos interesa ver el valor de la intensidad en el amortiguador con el valor de tensión recogido a la salida del operacional con la señal amplificada.

Hemos visto anteriormente que el carácter inductivo deforma la señal del PWM bastante, lo cual, para contrarrestar esta deformación en la señal pondremos un condensador detrás del amortiguador para obtener una señal más aceptable.



Los barridos a analizar fueron los siguientes:

## BARRIDO A UN 10 % DE DUTY CYCLE

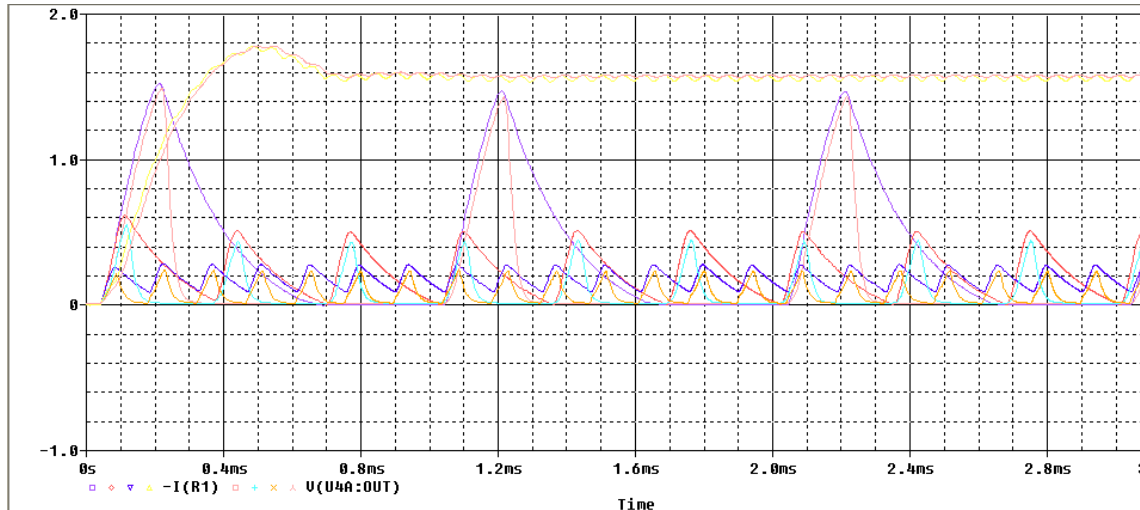


Figura 88: Duty cycle 10% con frecuencias de 1 KHz, 3 KHz, 7KHz y 15 KHz.

Aquí podemos observar, que pese a un porcentaje pequeño de duty cycle, las señales son bastante regulares a todas las frecuencias menos a las dos últimas a 7khz y 15 khz, por lo que estas dos últimas frecuencias las descartaríamos ya desde el inicio como frecuencias de trabajo de nuestro sistema.

## BARRIDO A UN 20 % DE DUTY CYCLE

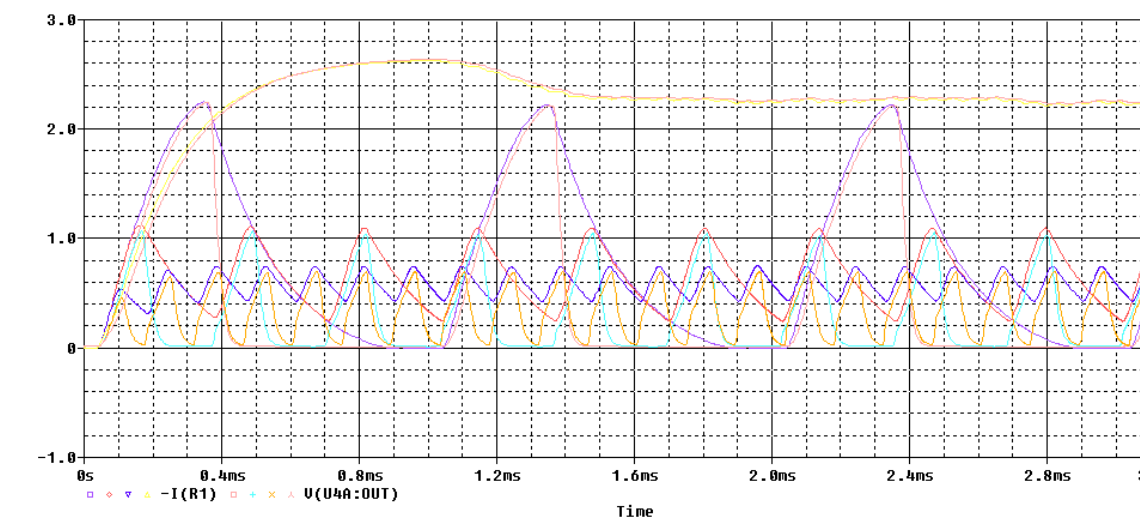


Figura 89: Duty cycle 20 % con frecuencias de 1 KHz, 3 KHz, 7KHz y 15 KHz.

Aquí en este barrido, vemos que las frecuencias de 1 khz y 3 khz son buenas para que nuestro sistema trabaje a estas, ya que ambas se comportan idénticas tanto en tensión como en corriente.

## BARRIDO A UN 40 % DE DUTY CYCLE

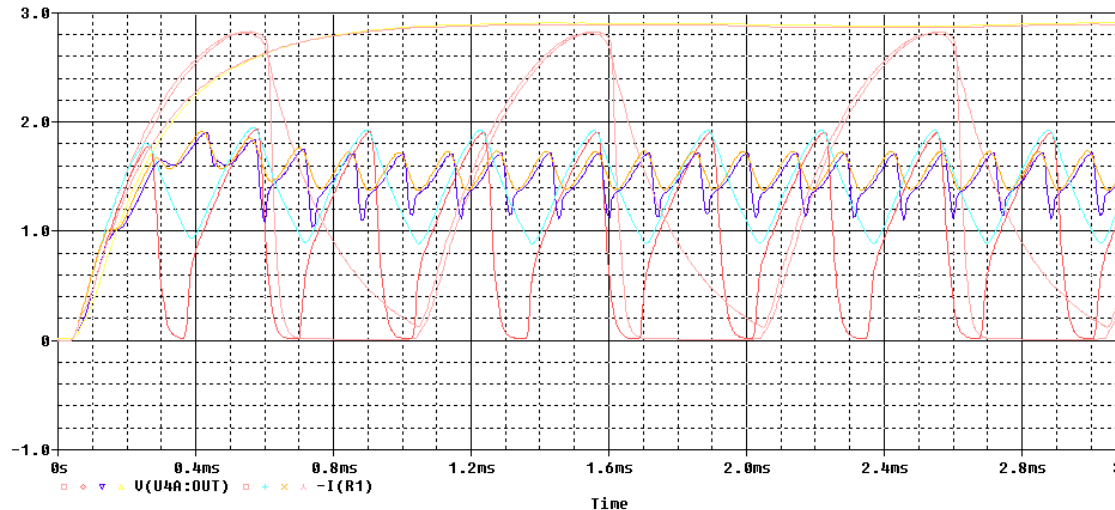


Figura 90: Duty cycle 40 % con frecuencias de 1 KHz, 3 KHz, 7KHz y 15 KHz.

En un barrido con un 40 % de Duty cycle, ya comenzamos a diferenciar que a 1 khz la señal es más fácil de trabajar que a 3 khz, por lo tanto ya comenzamos a decantarnos por la frecuencia a la que trabajara nuestro sistema.

## BARRIDO A UN 80 % DE DUTY CYCLE

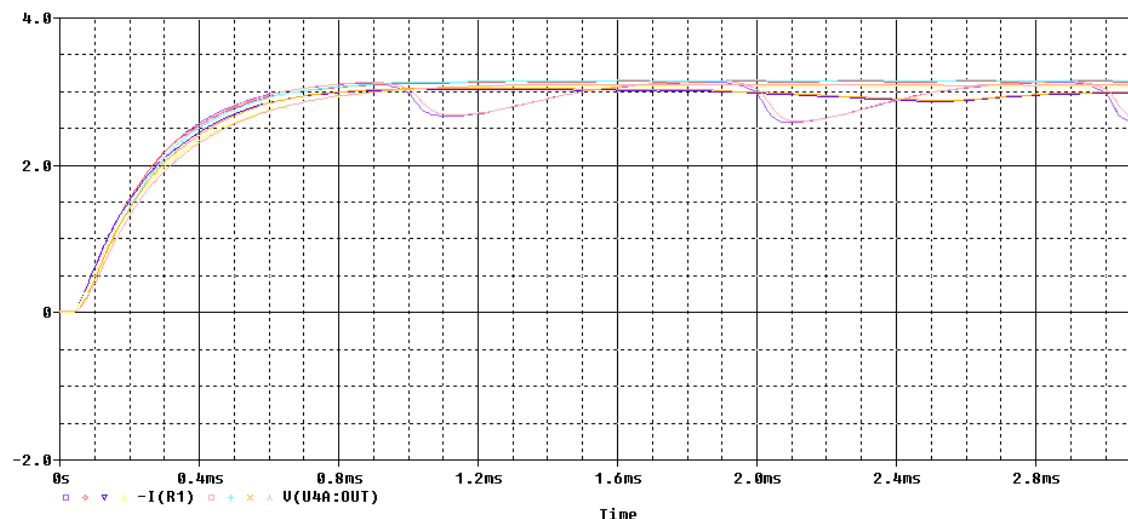


Figura 91: Duty cycle 80% con frecuencias de 1 KHz, 3 KHz, 7KHz y 15 KHz.

Aquí ya con un duty cycle muy grande, de un 80% vemos que la única señal tratable para el control por PWM es la de 1 khz, por ello esta será nuestra señal de trabajo.

Como conclusión de nuestro estudio es que a 1KHz de frecuencia en PWM nuestro sistema si podrá ser controlado por el ARM7.

Por último nos queda ver cómo nos mejora la señal el condensador de 100 nF que ponemos para mejorar la deformación de señal que hace el carácter inductivo del circuito.

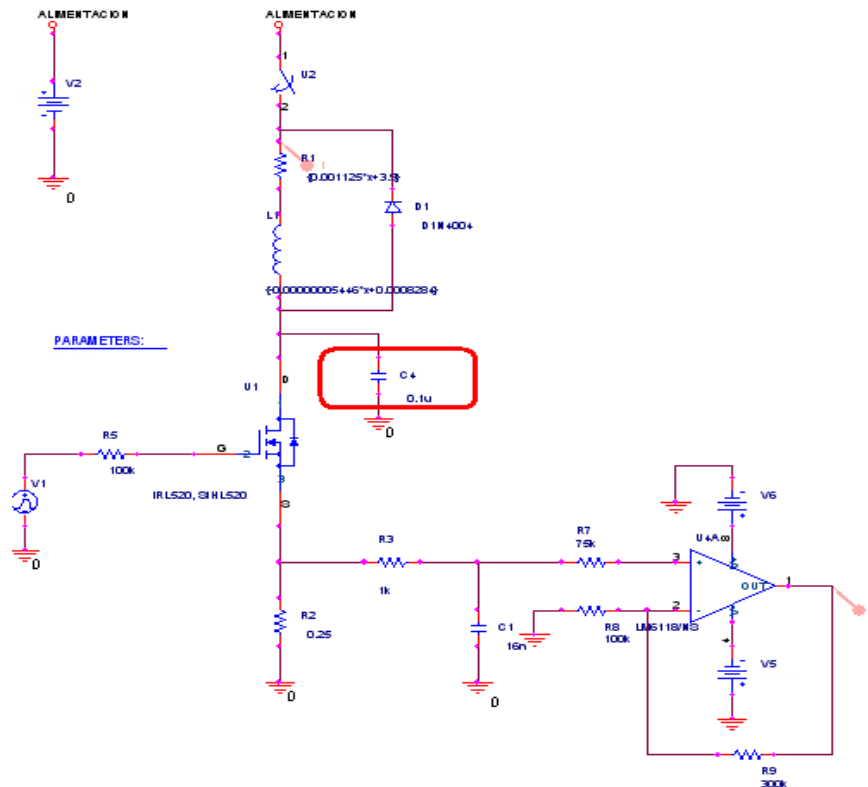


Figura 92: Esquema potencia mejorado

La señal que genera este cambio es la siguiente:

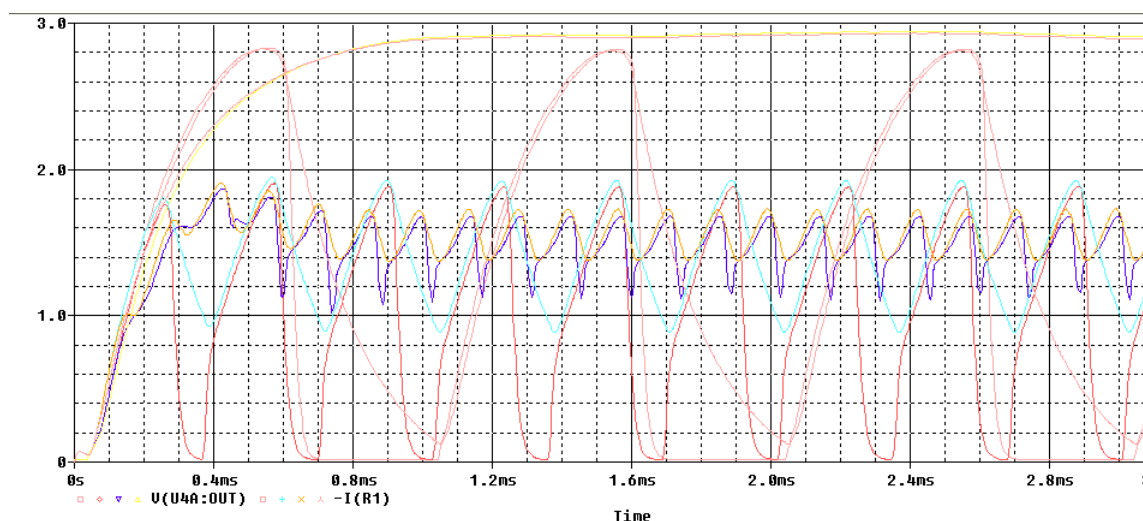


Figura 93: Señal PWM mejorada

Vemos que para un PWM del 40%, el condensador logra mejorar la señal en todas las frecuencias.

### 3.4.4. CONTROL PID (AJUSTE)

Las siglas PID hacen referencia al carácter de nuestro regulador. Este regulador tiene una parte proporcional, una parte derivativa y otra parte integral, y deberemos ajustar cada una de ellas para conseguir que la respuesta a nuestro sistema responda a unas especificaciones deseadas.

Nuestro proceso a ajustar mediante el PID será el siguiente.

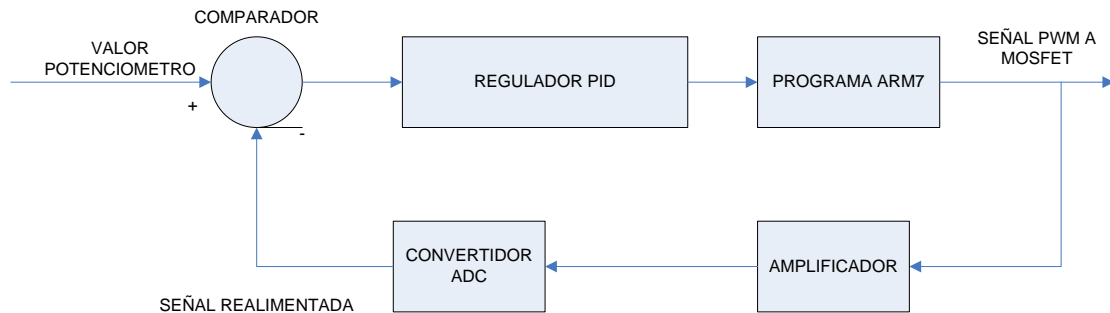


Figura 94: Lazo cerrado PID

Deberá cumplir la función en la cual el amortiguador no presente cambios bruscos cuando le variemos el valor de referencia en el potenciómetro.

En este ejemplo ilustrativo se muestra las distintas respuestas que puede generar un sistema ante un cambio en la señal, según la configuración del regulador PID

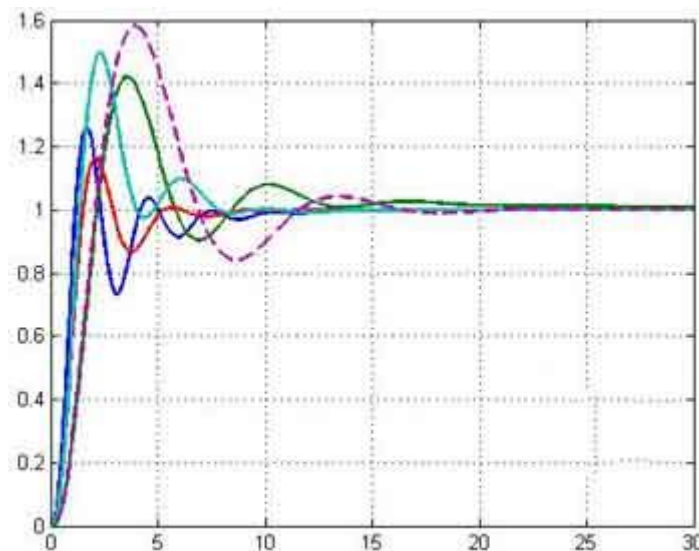


Figura 95: Respuestas PID

Es por ello, que ante todas las posibles respuestas buscamos la que más se ajuste a un funcionamiento suave del amortiguador.

La función PID implementada en el programa del ARM7 es la siguiente:

```
//definiciones del PID

#define epsilon 2    // epsilon es una cte
#define dt 0.001    //10ms es el tiempo de ciclo que tiene para dar la respuesta el PID
#define MAX 1023 //valor limite superior del PID
#define MIN 0      // valor limite inferior del PID
#define Kp 2       // cte de proporcionalidad del PID
#define Kd 0.0002  // cte de derivacion del PID
#define Ki 0.005   // cte de integracion del PID

....

//inicio variables PID

static float pre_error = 0;
static float integral = 0;
float error;
float derivative;
static int output;

.....

int PIDcal(int actual_position)
{

    //Calculamos el error entre el valor que marca el potenciómetro y el del T0MR1

    error = val_POT - actual_position;

    //Si el error es muy pequeño la parte integral no actua (<2)

    if(abs(error) > epsilon)
    {
        integral = integral + error*dt;
    }

    // Calculamos el valor derivativo del PID

    derivative = (error - pre_error)/dt;

    // Calculamos el valor nuevo del T0MR1 con valores del PID

    output = Kp*error + Ki*integral + Kd*derivative;
```

```
//Valores de saturacion del PID

if(output > MAX)
{
    output = MAX;
}

else if(output < MIN)
{
    output = MIN;
}

//Actualizamos el error para el siguiente ciclo

pre_error = error;

return output;
}
```

Observando la función PID, deberemos explicar los siguientes conceptos:

- Kp: es el valor que deberemos meter para ver la proporción en la cual varia nuestro sistema
- Ki: es el valor que deberemos meter para que nuestro sistema sea más integrador o menos
- Kd: es el valor que deberemos meter para que nuestro sistema sea más derivativo o menos.
- dt: es el tiempo que existe entre una medida que da el PID y la siguiente , y es conocido , pues es concretamente un ciclo de 1khz, que equivale a 0.001 sg.

Antiwindup: es una función que lo único que recoge, es que si regulando la señal por estos parámetros (proporcional, integral, derivativo) se nos sale de un rango deseado, el sistema le dará el valor límite de ese rango y borrara el que ha calculado fuera de él, esto nos sirve para no generar grandes errores a la hora del cálculo.

### **Como afecta estos parámetros a la regulación de la señal:**

El controlador más sencillo y efectivo que se usa es el control proporcional, que presenta una salida proporcional al error entre la salida deseada y la medida que el error disminuye, es decir, cuando la salida tiende al valor deseado, pierde “fuerza” ya que como el error tiende a cero la acción correctiva también tiende a cero y no se llega nunca al valor deseado de salida.

La acción Integral justamente corrige este defecto elevando el tipo de sistema y eliminando el error en el régimen permanente.

La acción derivativa mejora el transitorio y reduce el tiempo de establecimiento (el tiempo que tarda en llegar al valor deseado, o a un porcentaje del mismo).

En el siguiente cuadro se reducen estas características indicando que ocurre cuando aumentan los valores de  $K_p$ ,  $K_i$  y  $K_d$ .

	Sobre impulso	Tiempo de establecimiento	Error permanente
$K_p$	Aumenta	No afecta	Disminuye
$K_i$	Aumenta	Aumenta	Elimina
$K_d$	Disminuye	Disminuye	No afecta

El efecto Anti-Windup viene de que como la integral resulta de la sumatoria de los errores de cada ciclo, se corre el riesgo de saturar la integral para valores de error relativamente grandes, lo que trae aparejado oscilaciones bruscas y el problema de que al tener la sumatoria un valor grande se hace mayor el tiempo necesario para reducir la suma cuando se reduce el error.

Para evitar esto se hace trabajar a la integral solo dentro de valores de posición cercanos a la posición deseada de modo de integrar solo errores pequeños.

### 3.4.5. AJUSTE ADC ANCHO DE PULSO

Una vez que se ha implementado el programa, deberemos comprobar que nuestro sistema hace bien las conversiones de analógico a digital con el ADC, por ello realizamos una serie de cálculos previos para ver si se correspondían con los que nos daban las simulaciones en el programa de Keil:

Primero se observa que el máximo no es 3v sino 3,3 v, por lo que deberemos establecer la siguiente proporción:

ADC analógico  $\Rightarrow$  0-3,3 V  
ADC digital  $\Rightarrow$  0-1024 posiciones

Por lo tanto esta regla de tres simple nos valdrá para comprobar los cálculos:

3,3  $\Rightarrow$  1024  
2  $\Rightarrow$  x

Para este ejemplo de 2v, obtenemos que  $x = 620,6 = 620$

Ahora comprobamos con el ADC

Para 2v el ADCR marca:

ADCR = 0x00009B00

De los 32 bits que contiene este registro, el valor de la conversión siempre se guarda entre el bit 6 y el bit 15; por lo que cogemos el valor binario entre estos dos bits y nos da lo siguiente:

Valor = 1001101100; que pasado a decimal es 620;

Por lo tanto vemos que ha realizado bien la conversión.

Cogemos un segundo ejemplo:

Para 3v:

3,3  $\Rightarrow$  1024  
3  $\Rightarrow$  x

Nos sale que  $x = 930$

Y el valor que nos marca en el ADCR fue

ADCR = 0x0000E880

Cogemos el valor entre los bits 6 y 15:





Valor =  $1110100010 = 930$  en decimal, este segundo ejemplo también nos marca bien

Y en un tercer y último ejemplo en el cual acabar este test de funcionamiento cogemos el valor máximo:

El valor de 3,3 que equivale a 1024 posiciones

El ADCR nos marca:

ADCR =  $11111111 = 1023$ ,

Por lo tanto está bien, ya que son 1024 posiciones contando el cero.

## 3.4.6. DISEÑO EN PCB

### Diseño del Esquemático (Pasos)

La placa de circuito impreso se ha realizado con el programa OrCAD 9.

Con este programa se hace primero el esquema eléctrico y luego el mismo programa rutea la placa del circuito impreso.

Para comenzar un nuevo diseño en el OrCAD, se debe entrar en la sección de OrCAD Capture y realizar el siguiente pasó:

File -> New -> Project...

Introduciremos el nombre como desee que se llame nuestro proyecto y seleccionamos el nuevo tipo de proyecto que queremos realizar, que en nuestro caso PC Borrad Wizard.

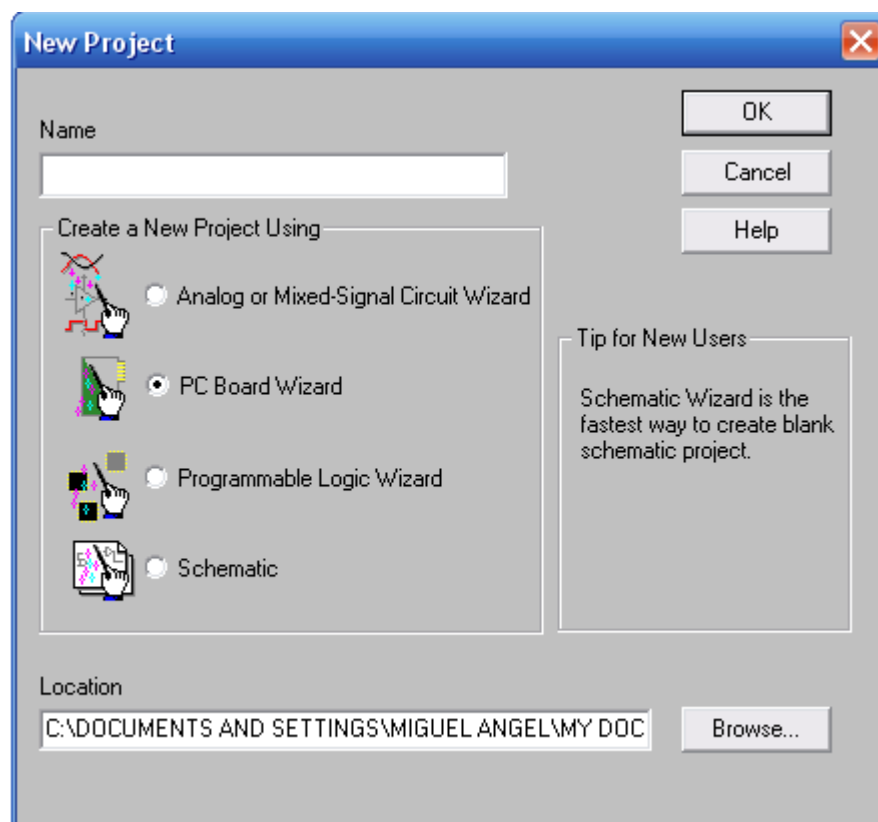


Figura 96: Configuración 1

Una vez realizado esto, debemos cargar los componentes que necesitemos de las librerías del OrCAD y seguidamente los uniremos con la opción wire que significa cable o hilo, y es la unión de los componentes de la forma que deseemos conectarlos, siguiendo el esquema eléctrico.

The diagram illustrates a PIR sensor circuit. The PIR sensor (P00M\_TIR) is connected to a 5V supply (J5) through a 100k resistor (R14). The sensor's output is connected to the non-inverting input (+IN A) of the LTI1490 op-amp (U2) through a 1.0k resistor (R15). The op-amp is configured with a voltage divider (R17, R13) at its non-inverting input (+IN A) and a feedback network (R2, R13) at its inverting input (-IN A). The op-amp's output (OUT A) is connected to the DUMPER\_12 output (J1) through a 100nF capacitor (C4). The DUMPER\_12 output is also connected to a 0.1uF capacitor (C1) and a 1k resistor (R18) leading to an LED (D4). The op-amp is powered by a 5V supply (J5) and a 0.25uF capacitor (C2). The op-amp's output (OUT A) is also connected to the DUMPER\_12 output (J1) through a 100nF capacitor (C4). The op-amp is powered by a 5V supply (J5) and a 0.25uF capacitor (C2). The op-amp's output (OUT A) is also connected to the DUMPER\_12 output (J1) through a 100nF capacitor (C4). The op-amp is powered by a 5V supply (J5) and a 0.25uF capacitor (C2).

The screenshot shows the OrCAD Capture application window with the title bar "OrCAD Capture - [PLACAFINAL.opj]". The menu bar includes File, Design, Edit, View, Tools, Accessories, Reports, Options, Window, and Help. The toolbar contains various icons for file operations, design tools, and simulation. The main workspace is titled "PCB". On the left, the "Design Resources" panel is open, showing a hierarchy of files. The "Library" folder is expanded, listing several component files from the OrCAD Capture library, including CAPSYM.OLB, Connector.olb, Misc.olb, MISC2.OLB, MiscLinear.olb, MiscPower.olb, PASSIVEFILTER.OLB, regulator.olb, and Transistor.olb. The "Outputs" folder is also visible, containing .placafinal.drc and .placafinal.mnl files. The "Referenced Projects" folder is empty.

Página 90

En esta pantalla tenemos todos los archivos que se crean y que hemos creado al realizar el proyecto.

El archivo PAGE1 (al que se le puede cambiar de nombre), es el archivo que contiene nuestro esquema eléctrico; este archivo nos muestra el esquema eléctrico haciendo doble clic en él.

Teniendo ya todo lo necesario en la parte Capture de OrCAD, debemos prepararlo para llevarlo a la parte del Layout.

Es en esta parte donde haremos el ruteado de la placa de Circuito Impreso.

Para pasar el esquema eléctrico al Layout, debemos seguir el siguiente camino:

Tools -> Annotate

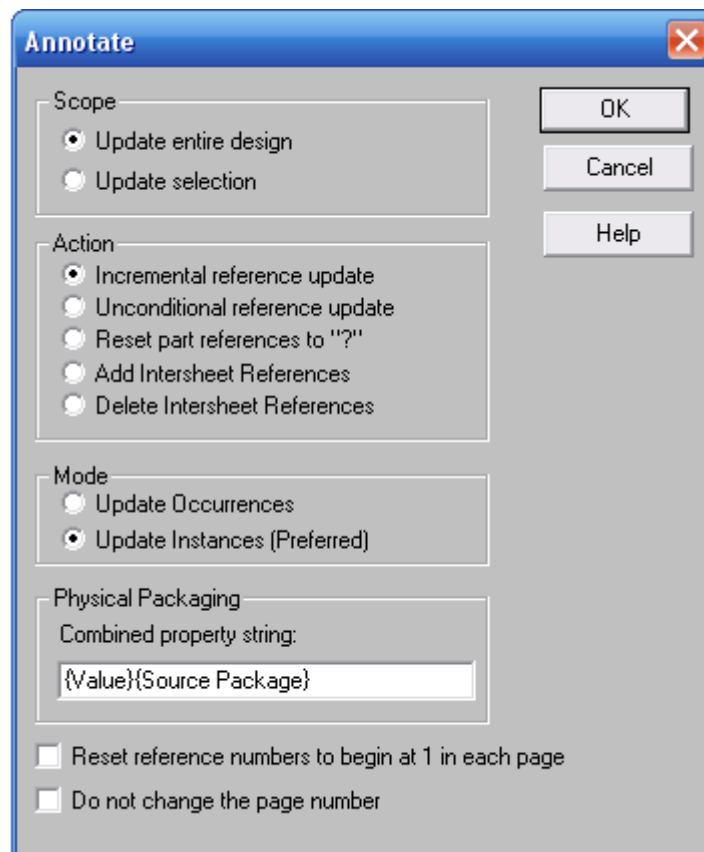


Figura 99: Configuración 3

Y de esta manera aceptaremos en estas condiciones y se nos generara un archivo con todas las referencias de componentes y conexiones que tiene nuestro circuito diseñado.

Después comprobaremos que nuestro circuito no contiene ningún error, para así generar bien los archivos que exportamos al layout, daremos por lo tanto:

Tools -> Design rules check

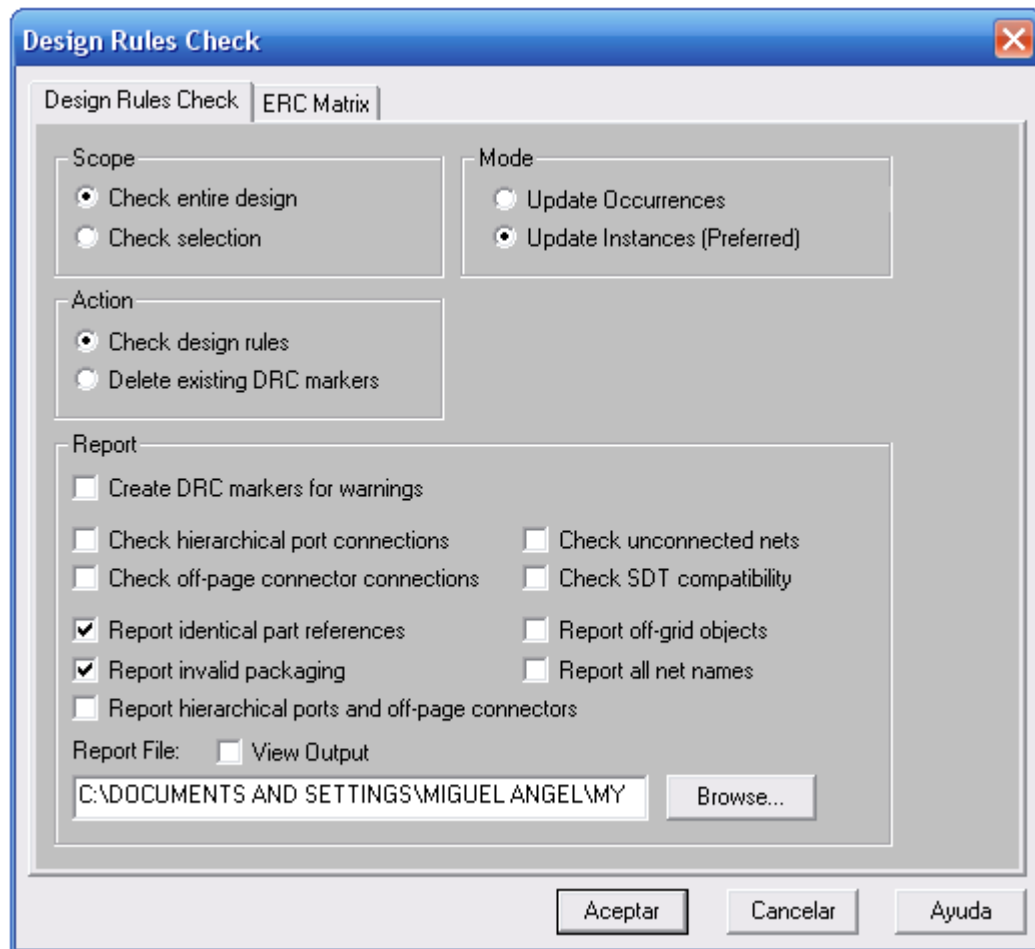


Figura 100: Configuración 4

Y mediante estas características el sistema nos comprobara si está bien diseñado o los posibles errores que hay.

Esta herramienta de detectar errores en el circuito, es más una ayuda que un elemento indispensable para detectar errores, ya que tendremos que tener total conocimiento de lo que hemos creado, y que es cada cosa, y mas como es en este caso haber creado partes e integrados, ya que ORCAD estos detalles no nos las distingue y menos si a la hora de crearla no lo hemos hecho detalladamente, como por ejemplo poner que era cada patilla (entrada, salida, pasiva). Así es deberemos clikear en ERC Matriz y configurar que queremos que ORCAD nos detecte y que nos lo indique como precaución (warning) o error así entraremos ahí y lo configuraremos:

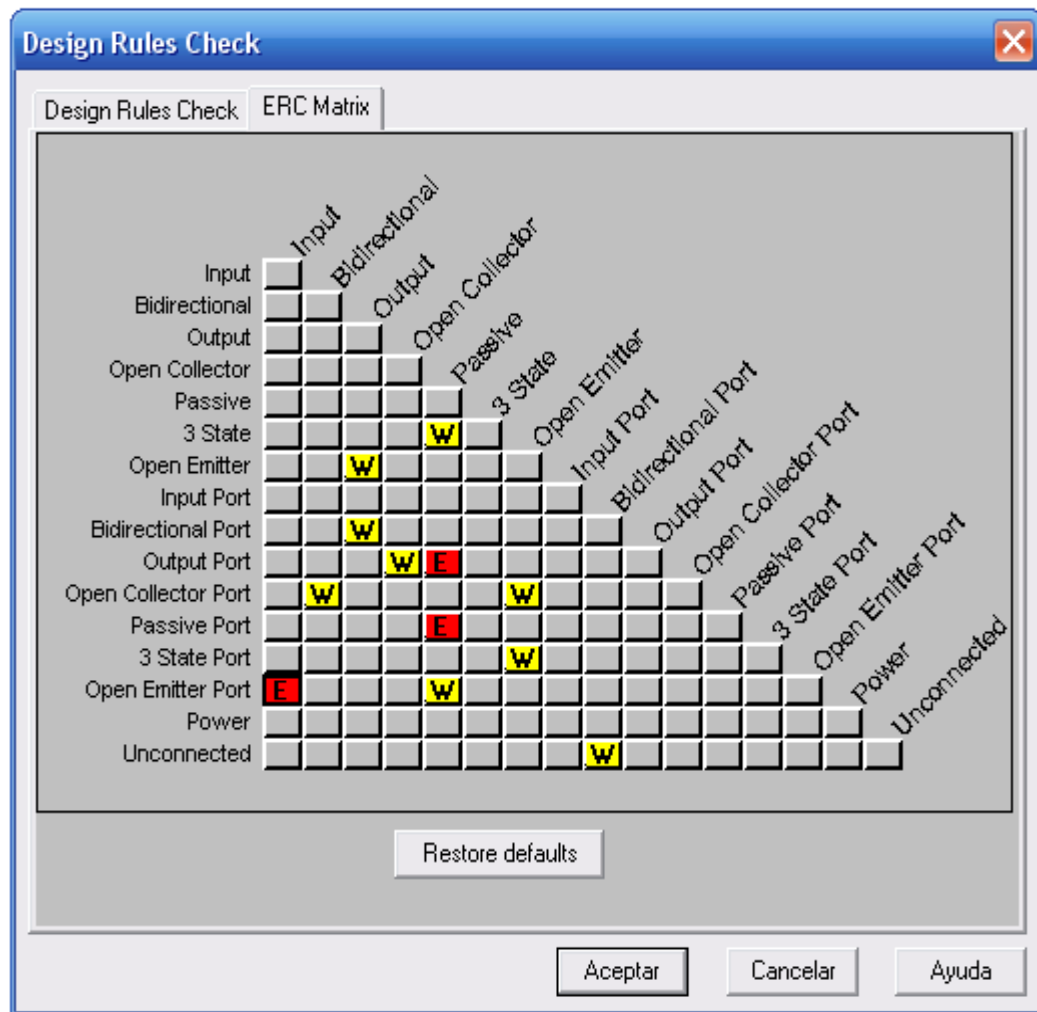


Figura 101: Configuración 5

Una vez comprobado esto ahora si daremos a:

Tools -> Create Netlist...

Dentro de aquí tendremos un submenú en el que nos aparecerán varias pestañas, de la que se elegirá la correspondiente a Layout, colocando las opciones como se ve en la figura:

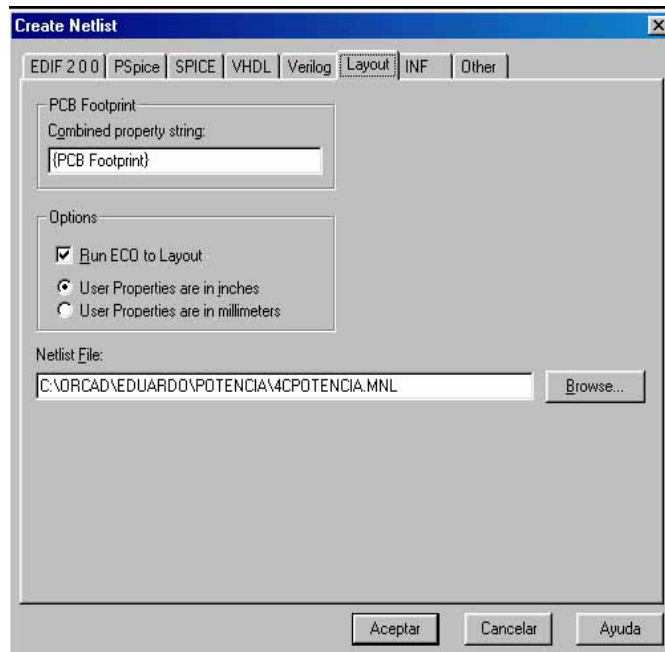


Figura 102: Configuración 6

Una vez hayamos realizado esto, pulsaremos el icono Aceptar, y seguidamente el programa OrCAD nos creará un fichero de extensión MNL, que podremos comprobar en el árbol principal del menú de OrCAD.

Este archivo llevará por defecto el nombre que le dimos al esquema eléctrico, que en nuestro caso será Placa\_final.mnl

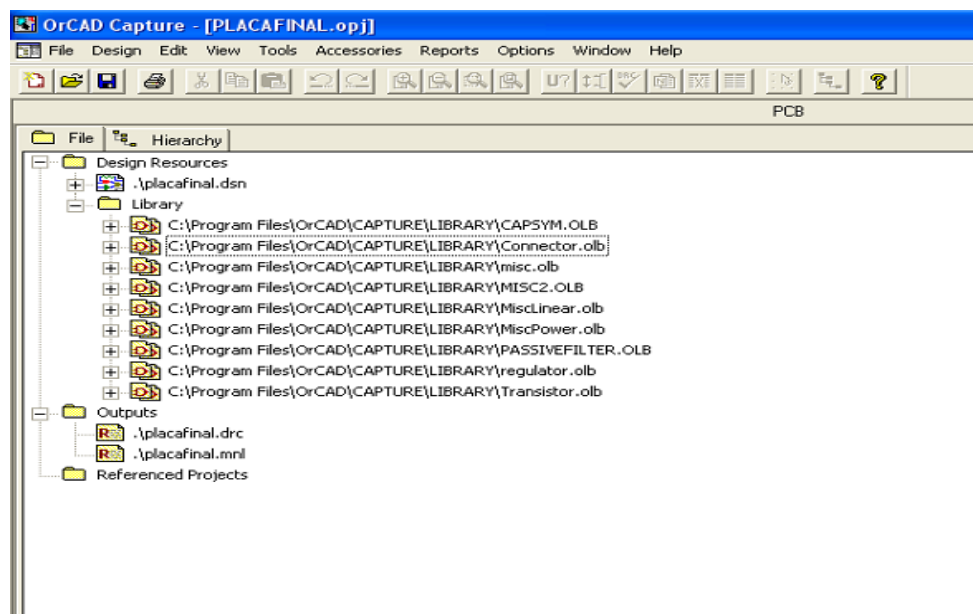


Figura 103: Configuración 7

En la figura de arriba se ve todos los archivos que se nos tienen que haber generado. Y ya está listo para pasarlo al layout.

## Diseño del PCB de la Placa (Pasos)

Abrimos un documento nuevo , y cargamos el archivo Default.tch, que está dentro de la ruta Orcad\Layout\_Plus\Data.

A continuación abrimos la netlist, el archivo con terminación \*.mnl, que hemos creado antes, y le indicamos donde guardar los archivos \*.max de layout. Si se recordara este archivo fue creado en el Capture para utilizarlo ahora en el layout.

Si hay algún error en los footprints al cargar la netlist nos aparecerá la siguiente pantalla, la cual sirve para relacionar el símbolo eléctrico con el símbolo topográfico. Para esto deberemos pinchar en **Link existing footprint to componente...** Si no encontramos el footprint adecuado a nuestro componente, este se puede crear el **LIBRARY MANAGER**, que esta explicado más adelante.

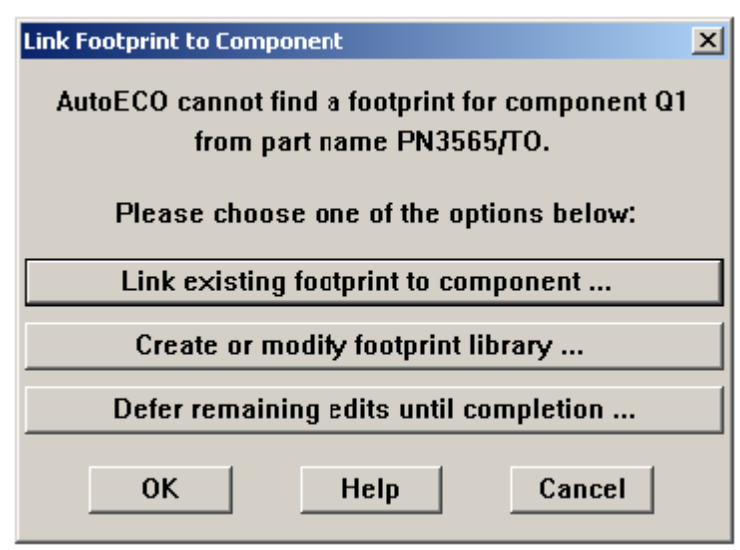


Figura 104:AutoECO

**Link existing footprint to componente:** Permite vincular un footprint existente al componente. Al seleccionar esta opción nos aparecerá esta pantalla, en ella debemos seleccionar la librería que queremos y el footprint que más se adapte al componente.



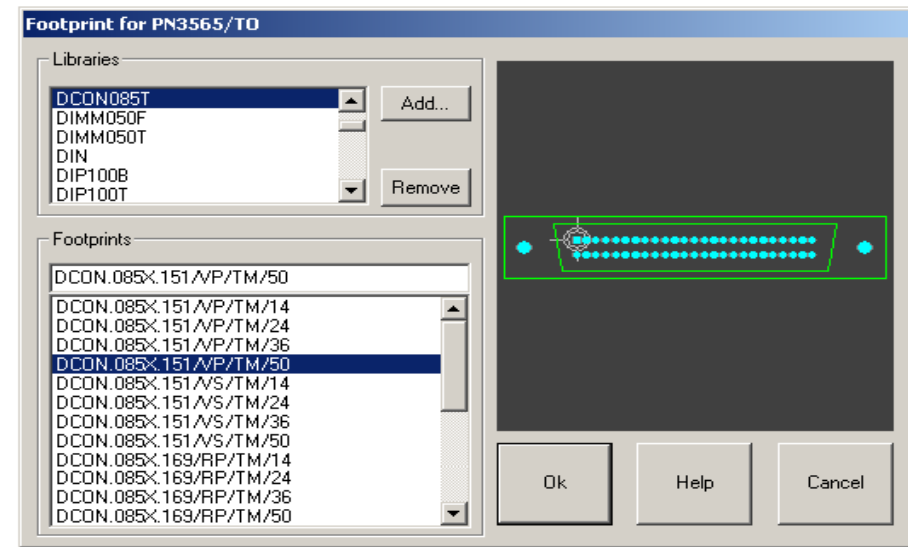


Figura 105: Footprint

**Create or modify footprints library:** Permite un nuevo footprint o modificar uno ya existente.

**Defer remaining edits until completion:** Retiene el resto de ediciones hasta que se ha completado el proceso AutoECO, mostrando al finalizar los errores encontrados en un fichero.

## Crear un footprint:

Pinchamos en el icono de **LIBRARY MANAGER**

En la nueva pantalla que aparece pincharemos en **CREATE NEW FOOTPRINT**.

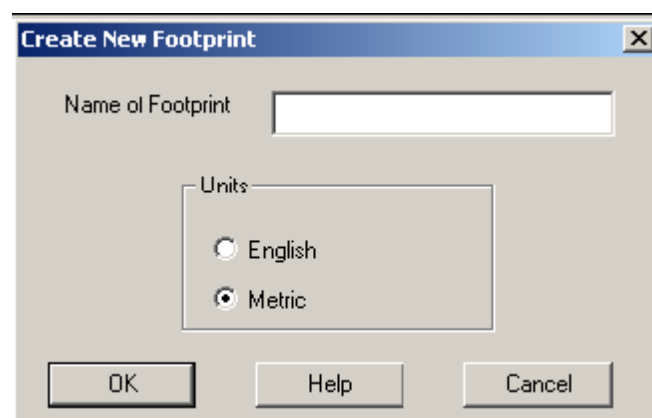


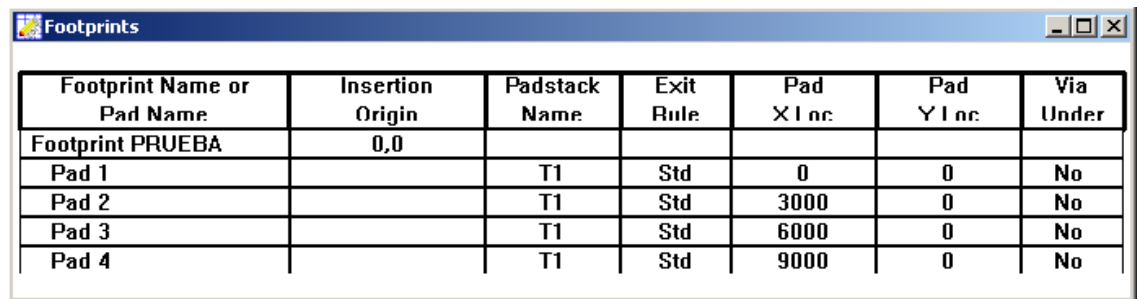
Figura 106: nombre footprint

Aquí introducimos el nombre y la unidad de medida que deseamos utilizar.

Ahora aparece una pantalla con un pin y una serie de etiquetas (referencia, nombre del componente, valor, nombre del encapsulado, etc), en esta pantalla será donde crearemos nuestro footprint.

Para colocar los pines deberemos pinchar en **PIN TOOL** y a continuación haremos botón derecho en el ratón y elegiremos **NEW**, y aparecerá un pin unido al cursor, para seguir insertando pines bastara con pulsar la tecla **INSERT** del teclado.

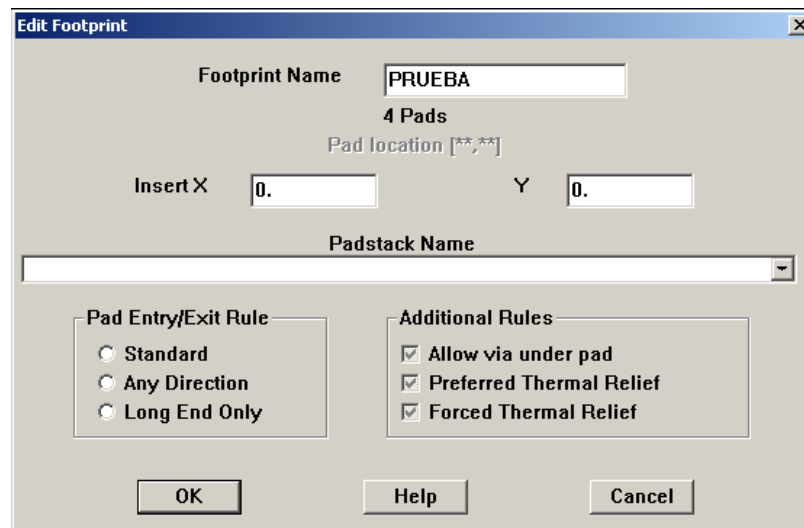
Para seleccionar una forma de nodo para todos los pines pinchamos en **VIEW SPREADSHEET, FOOTPRINTS**.



Footprint Name or Pad Name	Insertion Origin	Padstack Name	Exit Rule	Pad X Inc.	Pad Y Inc.	Via Under
Footprint PRUEBA	0,0					
Pad 1		T1	Std	0	0	No
Pad 2		T1	Std	3000	0	No
Pad 3		T1	Std	6000	0	No
Pad 4		T1	Std	9000	0	No

Figura 107: huellas componentes

Al hacer doble clic en el footprint se abre la ventana **EDIT FOOTPRINT**, y en el apartado de **PADSTACK NAME** definimos la forma del nodo.



**Edit Footprint**

Footprint Name: PRUEBA

4 Pads

Pad location [\*\*,\*\*]

Insert X: 0. Y: 0.

Padstack Name: [dropdown menu]

**Pad Entry/Exit Rule**

- ☐ Standard
- ☐ Any Direction
- ☐ Long End Only

**Additional Rules**

- ☒ Allow via under pad
- ☒ Preferred Thermal Relief
- ☒ Forced Thermal Relief

OK Help Cancel

Figura 108: definir nodo

Si lo que queremos es asignar un nodo a un solo pin, en la misma ventana de footprint que hemos abierto antes, en vez de hacer doble click en el footprint hacemos doble click en el pad que queremos modificar.

Ahora pasamos a dibujar el cuerpo del componente. Para ello pincharemos en el icono de **OBSTACLE** y haremos doble clic en la pantalla.

Aparecerá un cuadro en el que podemos definir el tipo de obstacle.

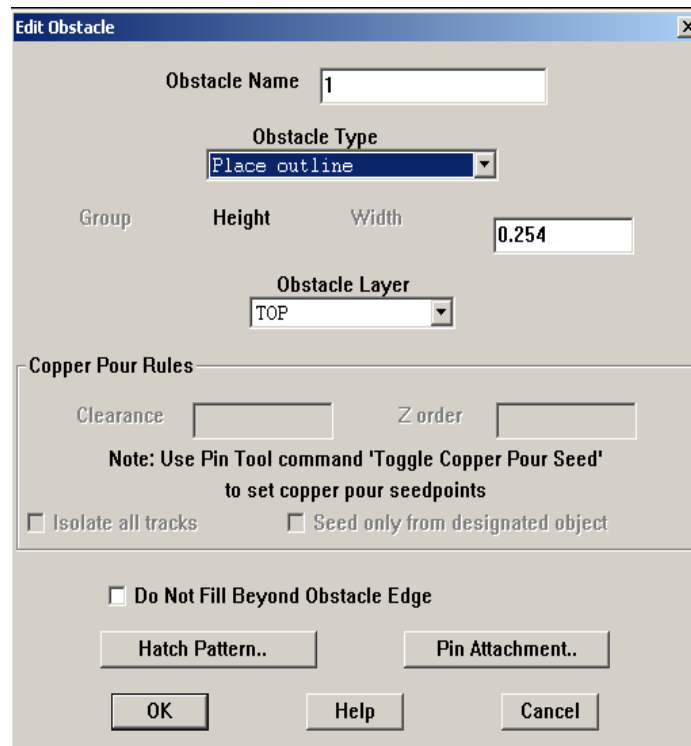


Figura 109: definición huella

Una vez terminado el componente, lo grabamos pinchando en el **SAVE** del **LIBRARY MANAGER**



Figura 110: guardar huella

Aparecerá otra pantalla en la que debemos indicar el nombre del componente y la librería en la que queremos guardarlo.

Una vez elegidos todos los footprint nos aparece nuestro diseño de la siguiente manera:

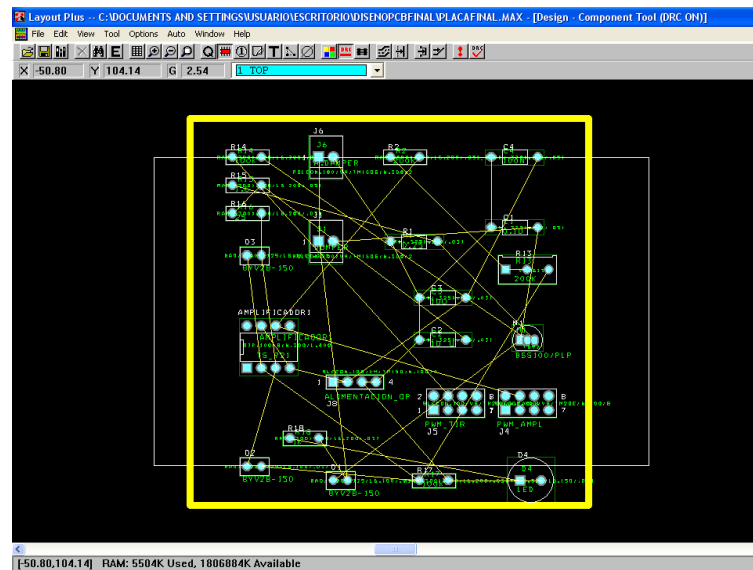


Figura 111: huellas PCB

En la barra de tareas tenemos distintos iconos para ir modificando nuestra placa.

Para comenzar a hacer las pistas indicamos a cuantas capas debemos hacerlo, es recomendable hacerlo siempre en la capa **TOP** y **BOTTOM** (que son superior e inferior respectivamente). Las capas **INNER1** e **INNER2** las desactivaremos al ser capas internas (en nuestro caso solo vamos a utilizar la capa superior).

Pinchamos en el botón de hojas de cálculo, en el menú que aparece, elegimos

**Layers** apareciendo la siguiente hoja de cálculo:

Layer Name	Layer Hotkey	Layer NickName	Layer Type	Mirror Layer
TOP	1	TOP	Unused	BOTTOM
BOTTOM	2	BOT	Routing	TOP
GND	3	GND	Plane	(None)
POWER	4	PWR	Plane	(None)
INNER1	5	IN1	Unused	(None)
INNER2	6	IN2	Unused	(None)
INNER3	7	IN3	Unused	(None)
INNER4	8	IN4	Unused	(None)
INNER5	9	IN5	Unused	(None)
INNER6	Ctrl + 0	IN6	Unused	(None)
INNER7	Ctrl + 1	IN7	Unused	(None)

Figura 112: Capas de pistas

Para que nos rute las pistas en una sola capa, debemos desactivar todas las demás capas, tanto las **Inner** como la **Top**. Esto lo haremos haciendo doble click sobre la capa que queremos desactivar, obteniendo un recuadro como éste:

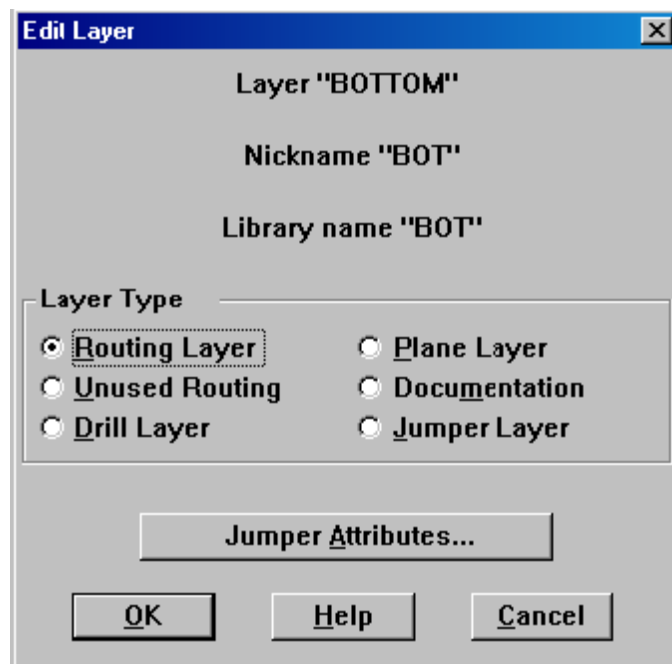


Figura 113: Creación ruteo pistas

Una vez colocados todos los componentes como se desea, probados que no hay ningún error al colocarlos, y configuradas las capas, es decir, **BOTTOM** que es la que utilizaremos en nuestro proyecto, seguimos con el proceso.

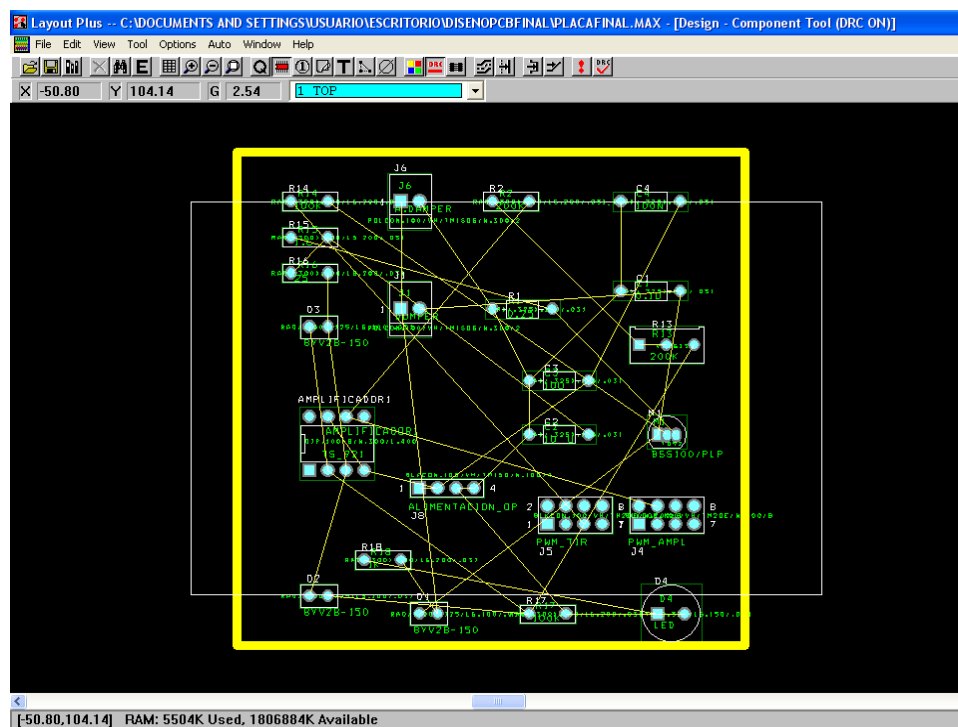


Figura 114: Placa PCB

Y por último, pinchamos en el menú de **Auto**, y elegimos **Autoroute Board** y automáticamente comenzara a rutear las pistas. . De forma análoga en **Auto Unroute Board** podemos deshacer el ruteado automático. Sobre todos debemos evitar que las pistas se crucen y que formen 90°, para ello es mejor ordenar los componentes antes de rutear. Una vez hecho esto quedara algo así:

Para definir el borde de la placa seleccionamos el botón, hacer click en las cuatro esquinas, y quedará definido el borde.

Una vez modificado todo y colocado, hemos ido cambiando las huellas según se ha creído conveniente, para ir de esta manera mejorándola poco a poco.

Para verificar los posibles errores que hemos cometido al ir diseñando la placa procederemos de la siguiente manera:

En el menú **Auto** seleccionamos la opción **Design Rule Check** apareciendo en la pantalla una ventana como esta.

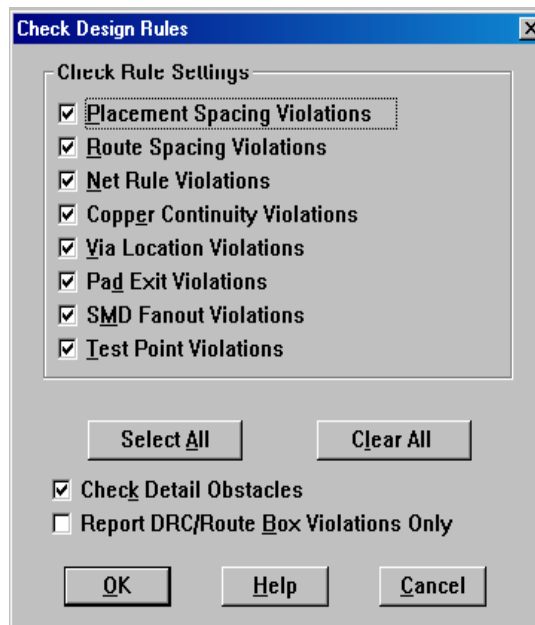


Figura 115: Check design rules

Y por ultimo para que nuestra placa no deje fallos de diseño, como por ejemplo ángulos de 90° o malas conexiones iremos a:

Auto>Cleanup Desing

Y la placa corregirá los errores que encuentre de este tipo, rehaciendo las pistas.

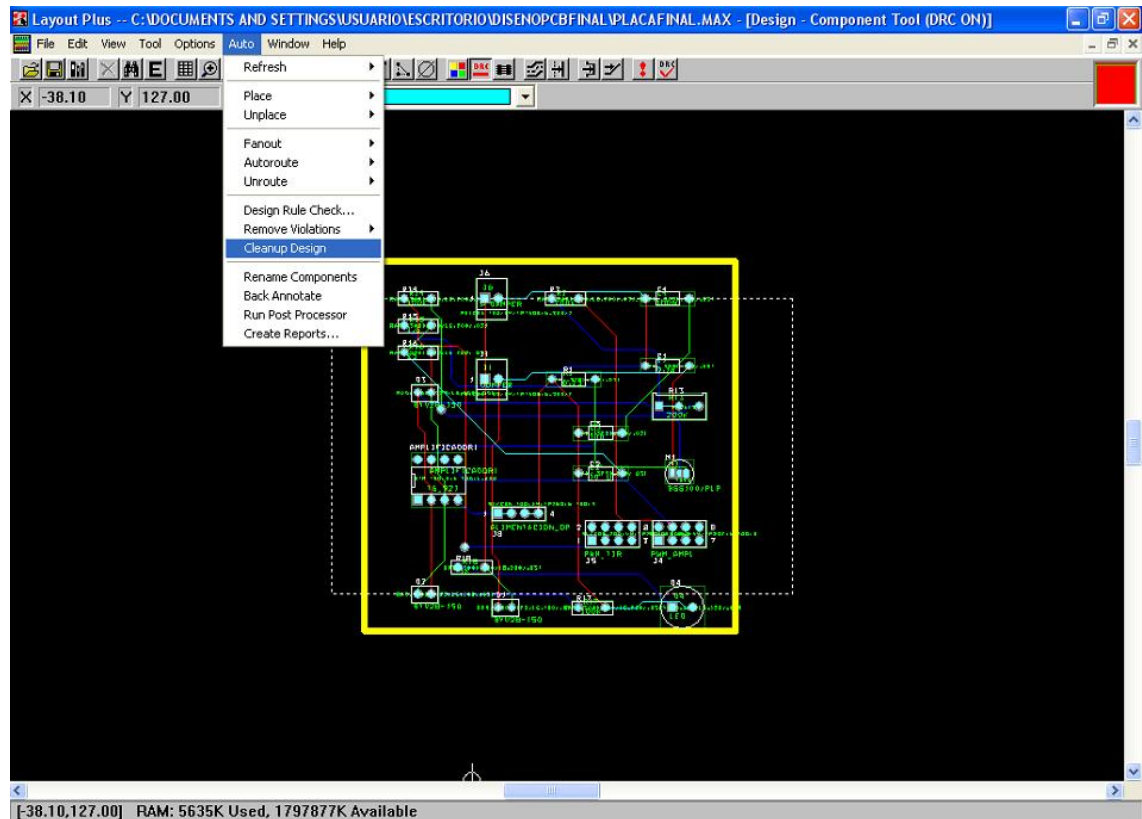


Figura 116: Corrección ruteado placa

Ahora cabe ver la magnitud de las pistas, dependiente de la intensidad que pase por ella, el diseño de grosor de pista será menor o mayor. Iremos al apartado nets, y ahí indicaremos el valor del grosor de pista.

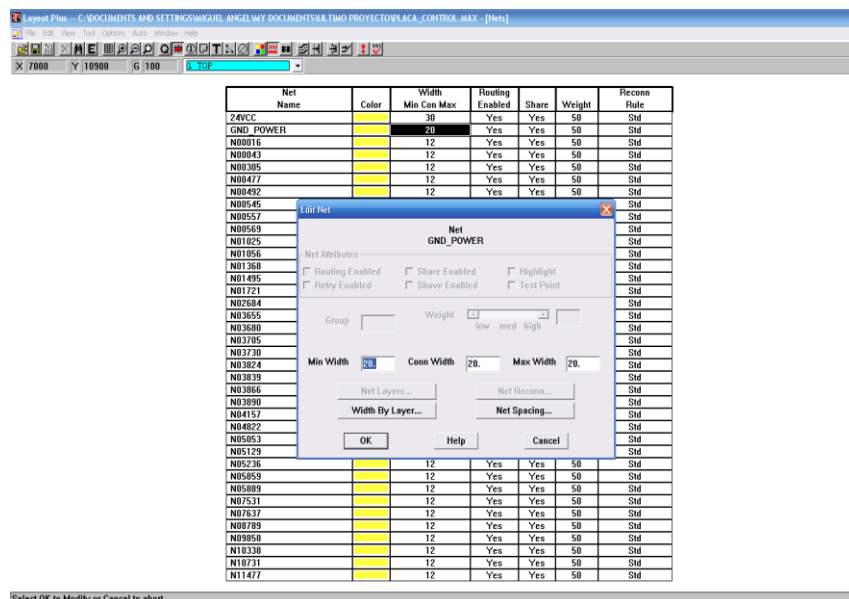


Figura 117: Modificación grosores de pista

Y el resultado final será este:

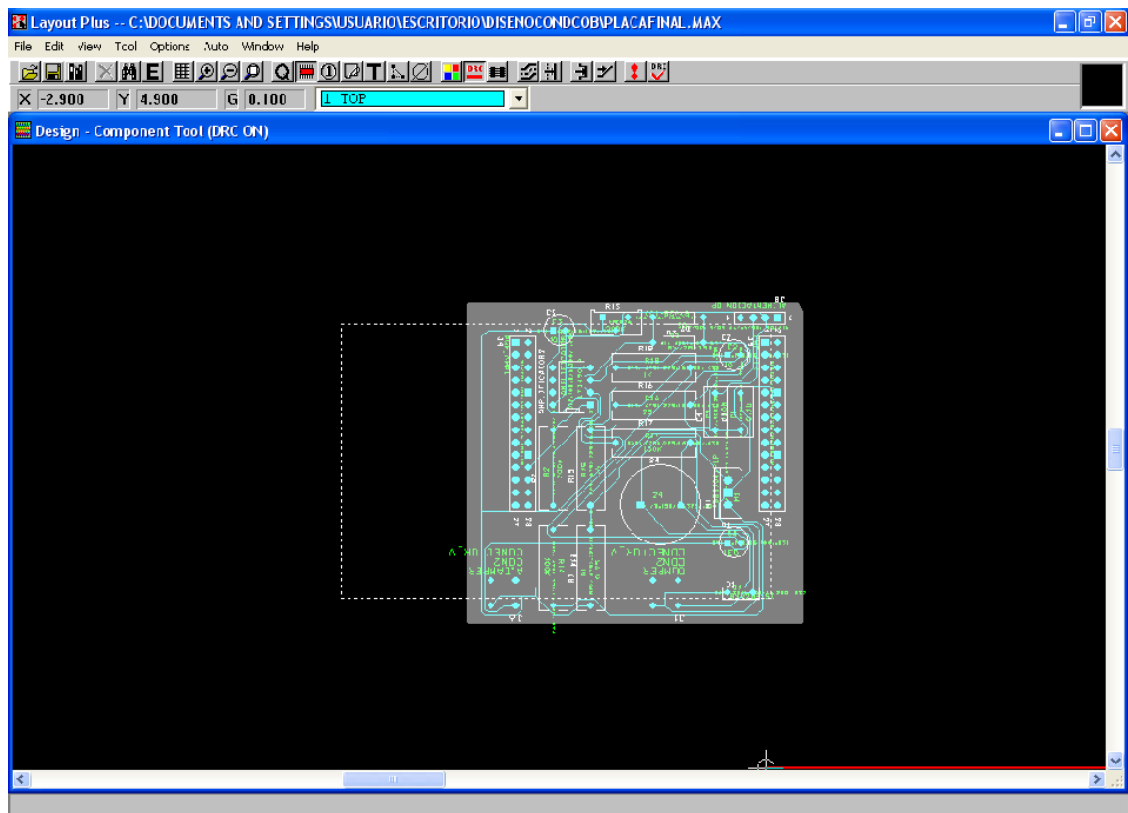


Figura 118: Ruteo acabado placa

Donde se puede observar el grosor de cada línea dependiendo de a que valla destinada.

Por último paso nos queda exportar los archivos a la maquina diseñadora de placas, pero antes debemos crear una serie de ficheros con distintas extensiones los cuales abriremos en CircuitCAM si fuese necesario.

Lo primero que tenemos que hacer es ir a **View**, y pinchar en **Database Spreadsheets**, y elegir **Post process** en el menú que nos aparece.

A continuación nos aparecerá un cuadro de datos, pinchando con el botón derecho aparecerá un menú emergente en el cual seleccionaremos **Run Batch**, el cual generará los archivos de forma automática.



## 4. PUESTA A PUNTO Y ENSAYO

### 4.1. INTRODUCCION

Para verificar y ajustar nuestro sistema electrónico diseñado para controlar a un amortiguador magneto-reológico, las pruebas que realizamos fueron las siguientes:

- En primer lugar probamos la placa con el canal AIN1 a 3v sacados de la propia placa y observar las conversiones que sacamos de la placa.
- En segundo lugar, probamos ya con la placa de potencia sobre el MCB2100 observando el funcionamiento y variando el potenciómetro
- Por ultimo lo que hacemos es ver las variaciones que se producen sobre el T0MR1 o duty cycle del sistema y ver que parámetros o constantes son las más apropiadas para una respuesta optima del PID.

### 4.2. PRUEBAS FUNCIONAMIENTO: TEST ANALITICO

Lo primero que hacemos es conectar todo el sistema, una vez hecho, lo que hacemos es sacar 3v de la placa y conectarlo al AIN1 del ADC por el P0.28, después conectamos el potenciómetro con el P0.27 que es el AIN0, y por último, en el P0.12 conectamos un led para ver si nuestro sistema parpadea por la interrupción del PWM generado.

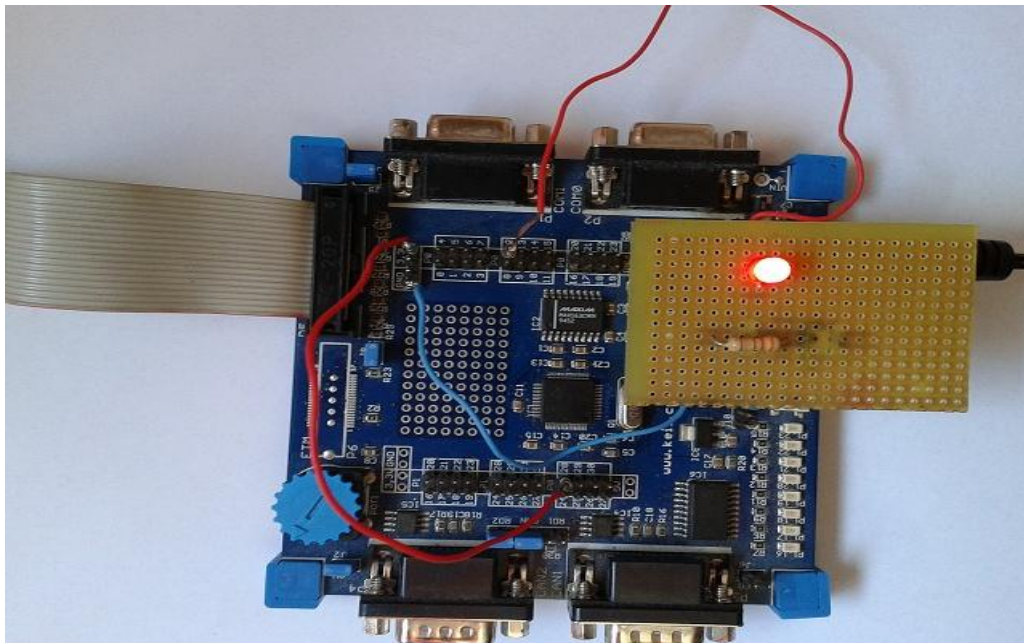


Figura 119:Placa trabajando en lazo abierto

## Proyecto Fin de Carrera

Después, lo que hacemos es establecer breakpoint en nuestro sistema:

Primero, establecemos dos break point en los flags de encendido y apagado y luego observamos si el led se apaga y se enciende, haciendo correctamente el PWM paso a paso.

Segundo, establecemos un breakpoint sobre el valor que vuelva el ADC en el T0MR1 y vemos si la conversión es la correcta para los 3 voltios a los que está conectado.

Y por último, y a la vez que miramos el valor del ADC, miramos también el valor del potenciómetro, lo variamos y vemos si está continuamente actualizando el valor del potenciómetro al valor que le modificamos físicamente.

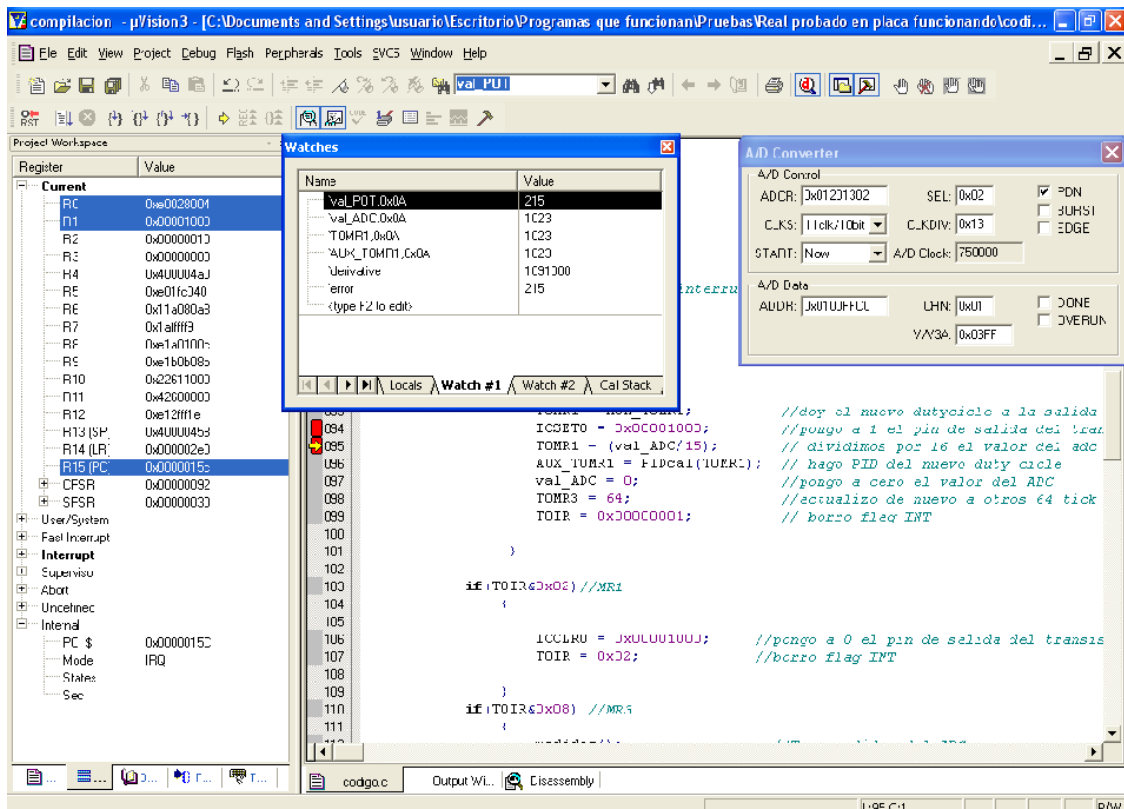


Figura120: Depurando programa en lazo abierto

## 4.3. PRUEBAS FUNCIONAMIENTO PID

Una vez que vemos que nuestro sistema funciona correctamente en los test, ahora probamos en lazo cerrado. Para ello, lo que hacemos es conectar nuestra MCB2100 con la placa de potencia diseñada para el amortiguador magneto-reológico. Una vez conectada variamos el potenciómetro y tomamos valores sobre la carga mediante un polímetro y vemos que efectivamente la tensión varía sobre esta, así como su tensión.

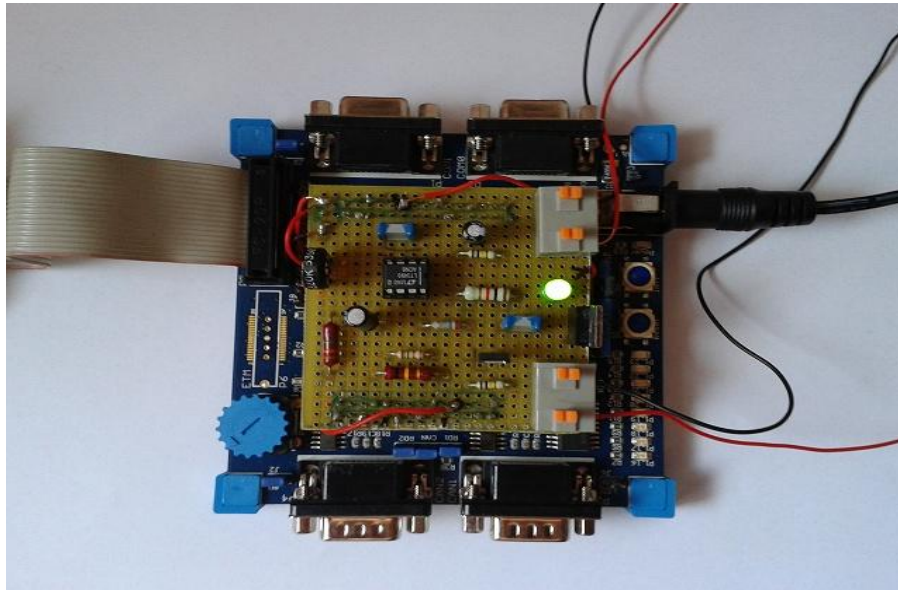


Figura 121: Placa trabajando en lazo cerrado.

Una última prueba, consiste en variar el potenciómetro y parar el programa y observar que valores nos marca el  $\mu$ Vision en ese momento sobre las variables de la placa

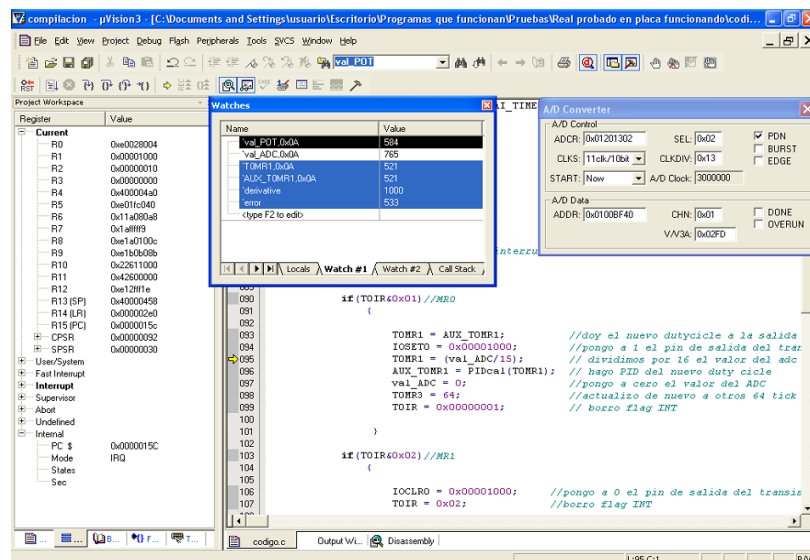


Figura 122: Depurando programa en lazo cerrado

Deberemos comprobar que el ADC va variando en función de los distintos anchos de pulso que tenemos y que el PID va por aproximaciones igualando el valor que le marca el potenciómetro de referencia.

### 4.4. AJUSTE DEL SISTEMA (GANANCIA)

Las conclusiones que hemos sacado de variar las constantes  $K_p$ ,  $K_i$  y  $K_d$  del PID, son las siguientes:

La variable  $K_p$  es la que más afecta, pues nos establece en mayor numero o menor número de ciclos el valor referencia a alcanzar, depende del valor que le demos, podemos decir que la rapidez con que conseguimos aproximarnos al número de referencia es mayor o menor.

La variable  $K_d$ , una vez realizadas pruebas, vemos que deberemos ajustarla para que haga las menores oscilaciones posibles sobre el valor referencia a alcanzar, es la segunda que más afecta al sistema.

La variable  $K_i$ , es la que menos afecta, pero es la que una vez establecido conseguimos mayor o menor oscilación en nuestro sistema, que este se “asiente” o no.

Como conclusión, debemos decir, que siempre que pongamos una carga diferente, en nuestro caso un amortiguador magneto-reológico diferente, deberemos ajustar los parámetros  $K_p$ ,  $K_i$  y  $K_d$  para conseguir que el sistema se estabilice y funcione bien, ya que cada amortiguador posee unos parámetros que varían y estos hacen variar a la señal, por lo que generan una señal distinta de PWM, y tendrá un ajuste diferente para el PID del programa.

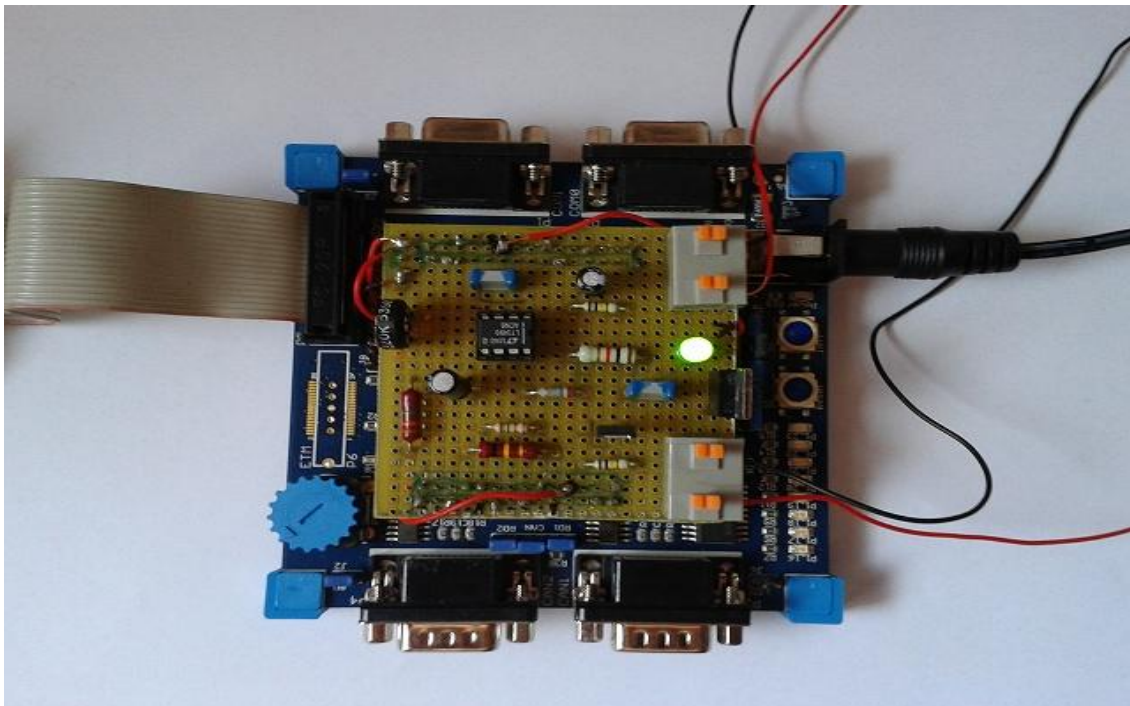


Figura 123: Probando sistema en funcionamiento.



### 5.CONCLUSIONES

En este Proyecto Final de Carrera podemos apreciar que se han tocado diversos campos de la ingeniería, entre ellos esta, la electrónica de potencia, circuitos eléctricos, electrónica digital, electrónica analógica y programación, así como varios conocimientos de mecánica para comprender el funcionamiento del amortiguador para su posterior tratamiento.

Pero a priori podemos definir tres grandes bloques de los cuales ha constado el desarrollo de todo este proyecto:

El primer bloque ha sido la caracterización, estudio y simulación del amortiguador y su comportamiento, así como la manera de adaptar el control a nuestra placa MCB2100. Ha sido una tarea ardua y prolongada de estudio y consulta de varios documentos e información, y varias simulaciones realizadas mediante la herramienta Pspice hasta conseguir caracterizar perfectamente nuestro amortiguador.

El segundo bloque ha constado del diseño primero mediante herramientas CAD como lo son Pspice y Layout de ORCAD y su posterior trabajo de ajustar y soldar los componentes físicamente a la placa diseñada y ruteada. En este bloque se ha llevado muchas horas de buscar los componentes adecuados, encontrar sus huellas y conseguir un espacio optimizado, ya que disponíamos de un espacio muy limitado para colocar la placa de potencia sobre la placa MCB2100, se resume en horas de ajuste y pruebas.

El tercer bloque ha sido el diseño del programa de control para el amortiguador, que para nosotros, ha sido uno de los más costosos, debido a la complejidad del microcontrolador, tanto en sus configuraciones como en su forma de programación del mismo. Desde el punto de vista del proyectante, este es en el que más se ha “sufrido” debido a la potencia del microcontrolador y sus grandes posibilidades de programación, que todo esto deriva en muchos registros a configurar y mucha capacidad de dominio de código de sistemas embebidos.

Como líneas futuras de mejoras podemos proponer varias:

- Crear una función que haga el autoajuste del PID de manera automática y así no tener que realizarla de manera manual.
- Conectar cuatro amortiguadores y conseguir que los programas se comuniquen entre ellos y varíen en sincronía todos en función de unos parámetros establecidos.
- Estudio de mejora de la señal del PWM, ya que el carácter inductivo del sistema la deforma mucho, en este Proyecto se ha intentado mejorar un poco, pero para conseguir una señal buena requiere un estudio en profundidad.

Como ultimo apunte a este apartado de conclusiones, se deberá indicar que la placa se realizo a mano soldando con el uso de una placa estándar con agujeros para realizar el sistema electrónico sobre ella. El motivo para la no realización de la placa sobre un PCB fue la falta de presupuesto, pero el diseño queda comprobado, verificado y guardado para futuras aplicaciones.

## 6.BIBLIOGRAFIA

- [1] Ingeniería de control, Modelado y control de sistemas dinámicos  
Luis Moreno, Santiago Garrido, Carlos Balaguer  
Editorial Ariel Ciencia
- [2] Circuitos Eléctricos, Sexta Edición  
James W. Nilsson, Susan A. Riedel  
Editorial Prentice Hall
- [3] Problemas de Electrónica de Potencia  
Andrés Barrado Bautista, Antonio Lázaro Blanco  
Editorial Prentice Hall
- [4] LPC2119/2129/2194/2292/2294 USER MANUAL  
Consultado en Marzo de 2011, disponible en web:  
[www.keil.com/support/man/docs](http://www.keil.com/support/man/docs)
- [5] The Insider's Guide to the Philips ARM7-Based Microcontrollers  
Consultado en Junio de 2011, disponible en web:  
[www.hitex.co.uk/arm](http://www.hitex.co.uk/arm)
- [6] Sistemas Electrónicos Digitales Basados en Microprocesador ARM  
Juan Vázquez Martínez, Michael García Lorenz, Juan P. García Nieto, Raúl Sánchez Reíllo  
Editorial Copy Red, S.A.
- [7] Ejercicios con ARM7 y entorno uVision  
Michael G. Lorenz, Juan Vázquez Martínez, Juan P. García Nieto, Raúl Sánchez Reíllo  
Editorial Copy Red, S.A.
- [8] [Alberto Rosa/Nano materiales y nano dispositivos]
- [9] [ [wikipedia.org](http://wikipedia.org) / sistemas en tiempo real]
- [10] [ [wikipedia.org](http://wikipedia.org) / Mosfet ]
- [11] [ [armandotripoides.com](http://armandotripoides.com) / amortiguadores ]

## ANEXO 1 PROGRAMA ARM7

```
#include <LPC21xx.H>
#include <stdio.h>

//definiciones del PID

#define epsilon 2           // epsilon es una cte
#define dt 0.001           //10ms es el tiempo de ciclo
#define MAX 1023           //valor limite superior del PID
#define MIN 0              // valor limite inferior del PID
#define Kp 2               // cte de proporcionalidad del PID
#define Kd 0.0002          // cte de derivacion del PID
#define Ki 0.005           // cte de integracion del PID

//configuracion general

    int val;
    int canal;
    static int val_ADC;
    static int val_POT;
    static int AUX_T0MR1;

//inicializo variables

    val=0;
    val_ADC=0;
    val_POT=0;
    AUX_T0MR1=0;

//inicio variables PID

    static float pre_error = 0;
    static float integral = 0;
    float error;
    float derivative;
    static int output;

//interrupciones

void RAI_TIMER(void) __irq; // RAI para TIMER 0
```

```
int main(void)
{

    //configuracion de los registros del ARM7

    VPBDIV = 0x00000001; // igualo cclk=pcclk
    PINSEL0 |= 0x00000800; // P0.12 como timer 0
    PINSEL1 |= 0x01400000; // habilito el AIN0 a P0.27 y AIN1 a P0.28
    IODIR0 |= 0x00001000; //configuro como salida el P0.12
    IOCLR0 = 0x00001000;
    T0PR = 0x0000003B; // preescalado a 59 tick (hecho con calculos) con 60 Mhz
    T0TCR = 0x00000002; // Reseteo counter y preescaler. Reseteo el contador del reloj
    T0MCR = 0x0000020B; // genero la interrupcion 0,1 y3 y pongo a cero el contador en MR0
    T0MR0 = 1023; // declaro el TMR0 a 1024 posiciones (tiempo de ciclo)
    T0MR1 = 600; // Duty cycle del PWM a 600 de un max de 1024
    T0MR3 = 64; // El ADC tomara valor cada 64 tick del ciclo

    T0TCR = 0x00000001; // habilitamos el timer

    //configuracion interrupcion del ciclo

    VICVectAddr0 = (unsigned) RAI_TIMER; // direccion de la RAI
    VICVectCntl0 = 0x00000024; // selecciona canal 4, slot 7
    VICIntEnable |= 0x00000010; // habilita INT

    while(1);

}

void RAI_TIMER(void) __irq //interrupcion del ciclo

{

    if(T0IR&0x01)//MR0
    {

        T0MR1 = AUX_T0MR1; //doy el nuevo dutycycle a la salida del transistor
        IOSET0 = 0x00001000; //pongo a 1 el pin de salida del transistor por el puerto P0.12
        T0MR1 = (val_ADC/15); // dividimos por 15 el valor del adc en el ciclo entero
        AUX_T0MR1 = PIDcal(T0MR1); // hago PID del nuevo duty cycle
        val_ADC = 0; //pongo a cero el valor del ADC
        T0MR3 = 64; //actualizo de nuevo a otros 64 tick la toma de datos del ADC
        T0IR = 0x00000001; // borro flag INT

    }

}
```



```
if(T0IR&0x02)//MR1
{

IOCLR0 = 0x00001000; //pongo a 0 el pin de salida del transistor por el puerto P0.12
T0IR = 0x02;          //borro flag INT

}

if(T0IR&0x08) //MR3
{
medidas();          //Tomo medidas del ADC
T0MR3 = T0MR3 + 64; // actualizo el ciclo del ADC en otros 64 tick mas
T0IR = 0x08;        //borro flag INT
}

VICVectAddr = 0x00000000; // borramos el contador de interrupciones
}

int PIDcal(int actual_position) // funcion del PID
{

//Calculamos el error entre el valor que marca el potenciómetro y el del T0MR1
error = val_POT - actual_position;

//Si el error es muy pequeño la parte integral no actua (<2)
if(abs(error) > epsilon)
{
integral = integral + error*dt;
}

// Calculamos el valor derivativo del PID
derivative = (error - pre_error)/dt;

// Calculamos el valor nuevo del T0MR1 con valores del PID
output = Kp*error + Ki*integral + Kd*derivative;

//Valores de saturacion del PID

if(output > MAX)
```

```
        {
            output = MAX;
        }

    else if(output < MIN)

        {
            output = MIN;
        }

    //Actualizamos el error para el siguiente ciclo

    pre_error = error;

    return output;

}

//Funcionamiento de mi ADC.

// Tomar medida por el canal seleccionado con el ADC

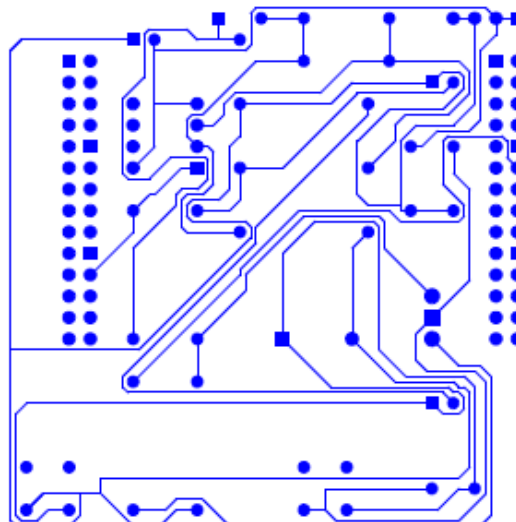
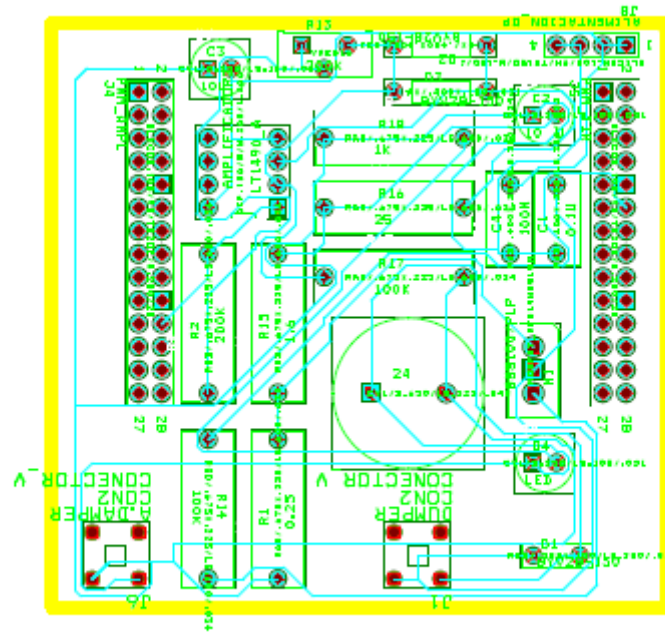
static unsigned short ADC_Read(unsigned char ch)
{
    unsigned int i;
    ADCR = 0x00201300 | ch;    // Inicio ADC y elijo canal
    ADCR |= 0x01000000;        // Empiezo A/D Conversion

    do
    {
        i = ADDR;                // Lee A/D
    } while ((i & 0x80000000) == 0); // Espero termine conversion
    return (i >> 6) & 0x03FF;    // bit 6:15 es bit 10 del AD
}

//Guardar valores de mi ADC

int medidas(void)
{
    val_POT = (ADC_Read(1));        // canal AIN0
    val_ADC = val_ADC + (ADC_Read(2)); // canal AIN1
}
```

## ANEXO 2 ESQUEMATICO PLACA



## ANEXO 3 PRESUPUESTO

Precio Mercado componentes Electrónicos					
Ítem	Cantidad	Parte	Unidades	Precio/Uds. €	Total Componente
1	1	zócalo LT1490A		0,62	0,62
2	2	condensador	10uF	0,26	0,52
3	2	condensador	100nF	0,37	0,74
4	1	diodo led	I=1ma	0,38	0,38
5	2	jumper 2 vías		0,3	0,6
6	2	jumper 20 vías	se ajustara a necesidades de la placa	3,55	7,1
7	1	Resistencia	0.25 ohm 5w	3,02	3,02
8	1	Resistencia	200k 0.25 w	0,89	0,89
9	2	Resistencia	100k 1w	1,05	2,1
10	1	Resistencia	1,5 ohm 600mw	0,012	0,012
11	1	Resistencia	75 ohm 3w	1,6	1,6
12	1	Resistencia	1k 3w	1,77	1,77
13	1	Amp. LT1490A		3,39	3,39
14	1	Mosfet RFD14N05L		0,87	0,87
15	1	diodo potencia BYW29		0,79	0,79
16	2	diodo NI1007		0,21	0,42
17	1	Potenciometro	200 k	1,2	1,2
18	1	cable soldar negro	rollo 3 metros	2,05	2,05
19	1	cable soldar amarillo	rollo 3 metros	2,05	2,05
20	1	estaño	rollo 2 metros	2,5	2,5
				Total	32,622

Servicios usados			
Servicio	horas	Precio/horas €	total €
Servicios de software (Orcad, Keil, Office)	180	100	18000
Soldador	40	15	600
Laboratorio	80	30	2400
herramienta especial (tenacillas, pinzas, cutter, polímetro)	80	5	400
total			21400

PERSONAL					
Apellidos y nombre	N.I.F.	Categoría	Observaciones	Dedicación total horas	Coste (Euro)
Torres Corrochano, Miguel Ángel		Ingeniero Junior	Responsable de Proyecto	1760	15714,28571
García Lorenz, Michael V.		Ingeniero Sénior	Coordinador de Proyecto	480	7142,857143
				Total	22857,14286

SUBCONTRATACIÓN DE TAREAS		
Descripción	Empresa	Coste Computable (€)
Caracterización Amortiguador	Departamento Tecnología Electrónica	3000
Total		3000

Presupuesto Costes Totales	Presupuesto Costes Totales(€)
Componentes	33
Servicios	21.400
Personal	22.857
Subcontratación de tareas	3.000
Total	47.290